Mathematical Biosciences and Engineering

MBE, 18(1): 712–726.
DOI: 10.3934/mbe.2021038
Received: 12 October 2020
Accepted: 07 December 2020
Published: 18 December 2020

http://www.aimspress.com/journal/MBE

Research article

An improved polychromatic graphs-based BOM multi-view management and version control method for complex products

Jun Chen[1,2], Gangfeng Wang[2,*], Tao Xue[1] and Tao Li[2]

[1] School of Computer Science, Xi'an Polytechnic University, Xi'an 710048, China
[2] Key Laboratory of Expressway Construction Machinery of Shaanxi Province, School of Construction Machinery, Chang'an University, Xi'an 710064, China

* **Correspondence:** Email: wanggf@chd.edu.cn.

Abstract: With the increasing complexity of modern engineering and wide application of intelligent manufacturing technology, the type and quantity of product data continue to climb, and the corresponding version iteration rate continues to accelerate. In order to achieve effective unified management and maintenance of product data and to track version evolution accurately, this paper proposes an improved polychromatic graphs-based BOM multi-view management and version control method for complex products. Firstly, the framework was established for BOM multi-view management and version control, whose mechanism was also elaborated. According to the model of multi-view-version product structure, the expression method of product structure was discussed at different stages of product life cycle. Furtherly, with multi-object constraints sets were introduced, a version control model was established based on the improved polychromatic graphs, whose definition and principle were also detailed. Finally, application cases were discussed to verify the feasibility and universality of the proposed models and methods.

Keywords: smart manufacturing; BOM; improved polychromatic graphs; multi-view management

1. Introduction

The bill of material (BOM) [1,2] is the main content and important data subset of the Digital Product Definition (DPD), also called the product structure table/product structure tree, which describes the structured parts of the product table, including a list of all sub-assemblies, parts, raw materials, and the attributes of materials required to make an assembly.

It laid the foundation for a new type of production organization and made the production management of manufacturing enterprises more scientific. With the development of information technology and intelligent manufacturing [3], the concept of BOM extends to the full life cycle of the product, which not only improves the transparency of design, process and manufacturing data, but also greatly improves the seamless integration and sharing of enterprise data.

At different stages of the product life cycle, the product structure will have a corresponding BOM type [4]. The most basic forms are design BOM, process BOM and manufacturing BOM. On the basis of these BOMs, quality BOM, packing BOM, maintenance BOM, etc. are also derived. With the development and application of intelligent manufacturing, BOM has become the most core part of modern manufacturing engineering.

Currently, companies use BOM to manage the hierarchical relationship of parts/components and their associated product data. However, the corresponding BOM in the product life cycle is usually separated in each digital manufacturing system, and unified management is not well realized. As a result, it is difficult to effectively maintain the consistency, accuracy, and validity of various product data and their relationships.

What's more, the product structure is becoming more complex, and engineering changes in the design, manufacturing, and maintenance processes are also becoming frequent, all of these variations have accelerated BOM change. So, characteristics of BOM's diversity, dynamics and relevance are more obvious, especially when there are multiple object association constraints, the complexity of BOM becomes much higher. Due to the fragmentation of BOM information, the management, it also will make a huge impact on maintenance and traceability of product BOM versions.

Therefore, this paper proposes a BOM multi-view management and version control method for complex products. The research contents are as follows: Section 2 summarized and analyzed the current BOM and version control research work; section 3 constructed a complex product BOM multi-view management and version control architecture; section 4, the key methods and technologies in the framework were elaborated; section 5 verified the proposed method through application examples; finally, the paper summarized and prospected.

2. Related work

With the fiercer market competition and growing diversification of products, the product structure has become increasing complex and the product listing cycle has gradually shortened. As a key link of production and operation for enterprises, BOM information has been particularly important. If the enterprise wanted to achieve collaborative work efficiently, it would be necessary to ensure the completeness, consistency and accuracy of BOM.

Although the application of information technology has facilitated BOM management to a certain extent, it's not easy to achieve unified management. First of all, BOM information is usually split in various information systems. Furthermore, BOM change is becoming more and more frequent. Moreover, there are lots of relationships of associations and constraints among versions. These would make enterprises confront with tremendous pressure on management of BOM information as well as BOM version control. Therefore, in response to above problems, many studies have been carried out as follows:

2.1. BOM management

A BOM is a list of all parts, ingredients or materials needed to make one product, which may also be called a formula, recipe or ingredients. The data organization and representation style for a BOM is called the structure of the BOM or the product structure [5]. With the research of BOM management, BOM has gradually evolved from a single narrow concept to various forms, forming a full life cycle BOM structure.

Liu et al. [4] and Huang et al. [6] proposed an XBOM management plan for the full life cycle of a product, which defined the BOM as design BOM, process BOM, manufacturing BOM, procurement BOM, cost BOM, and quality BOM. Etc. Combining information technology with green manufacturing theory, Xiang et al. [7] proposed the concept of energy consumption bill of materials (ECBOM), which is similar to manufacturing BOM (MBOM) or other extended BOM for different business scenarios. Zhou et al. [8] developed static service BOM (SBOM) for complex products to effectively organize product data for providing satisfactory maintenance, repair and overhaul service.

Cai et al. [9] proposed mapping rules of BOM multi-view model, and the mapping algorithm of BOM multi-view model was designed and implemented. Xia et al. [10] proposed a mapping method for BOM multi-view, which could record all the mapping parameters of all kinds of BOM through transforming-table to accomplish mutual transformation among different BOM views.

2.2. BOM version control

With rising complexity, dynamics and diversity of modern product, engineering change is more frequently, which leads to the continuous acceleration of BOM version change. In order to accurately record the entire process of BOM version change as well as to achieve precise traceability when problems occur, BOM version control has become increasingly important, and corresponding researches have been constantly carried out.

Li et al. [11] built product structure information tree version model, which supported a set of rules to control the version of product structure information tree based on version control lock. Zhang et al. [12] proposed a multi-view tree version management model and designed the version mechanism of BOM evolution and maintenance. Liu et al. [13] presented a definition of incremental version based on directed acyclic graph to improve search efficiency and storage performance of multi-version in collaborative design. Chen et al. [14] proposed a new multi-version management model based on the theory of polychromatic sets to reduce the storage space of multi-versions.

2.3. Summary

With development of information technology and the proposal of intelligent manufacturing, BOM has performed growing key role in the production and manufacturing.

Product data is gradually organized and managed through BOM, the concept of the full life cycle BOM was proposed to form various BOM type such as design BOM, process BOM, manufacturing BOM and so on, which can describe and manage the product data of corresponding life cycle stages. However, how to manage these BOM types in a unified manner to ensure the consistency, effectiveness and completeness of product data in the life cycle requires further research.

Some models of BOM version control had also been proposed, which can be summarized into linear model, tree model, directed acyclic graph model and polychromatic graphs model. In these, linear model focuses on recording change order of versions, but difference between versions cannot be expressed well; Tree model has a clear structure, yet it's difficult to express version merging; Although directed acyclic graph model solves problem of version merging, it's not easy to express relationship between versions; Polychromatic graphs model combines the advantages of models mentioned, however, it is not distinct to describe relationship between versions of different objects.

Therefore, based on above research, the paper proposed the corresponding framework, model and method for BOM multi-view management and version control of complex products, which would be elaborated as follows.

3. The framework of BOM multi-view-version management for complex products

Product data come from different stages of design, process, and manufacturing which can serve the corresponding business process with associated product structure. As mentioned earlier, at each stage of product life cycle, the product structure will have a corresponding BOM type. As a result, different users can create, use and maintain their domain data using BOM according to their own business requirements.

Although product data in various fields may exist in different systems, the corresponding BOM can organize these data into a logically single data storage entity, and establish strict constraints to form SSPD (single source of product data) [15,16]. Furthermore, it cooperates with a reasonable version control mechanism to ensure that the product data is consistent, complete, redundant and reliable. Besides, it's necessary to enable business members in various fields to independently access and use product data within their own authority. Therefore, this paper constructs a complex product multi-view BOM management model, as shown in Figure 1.

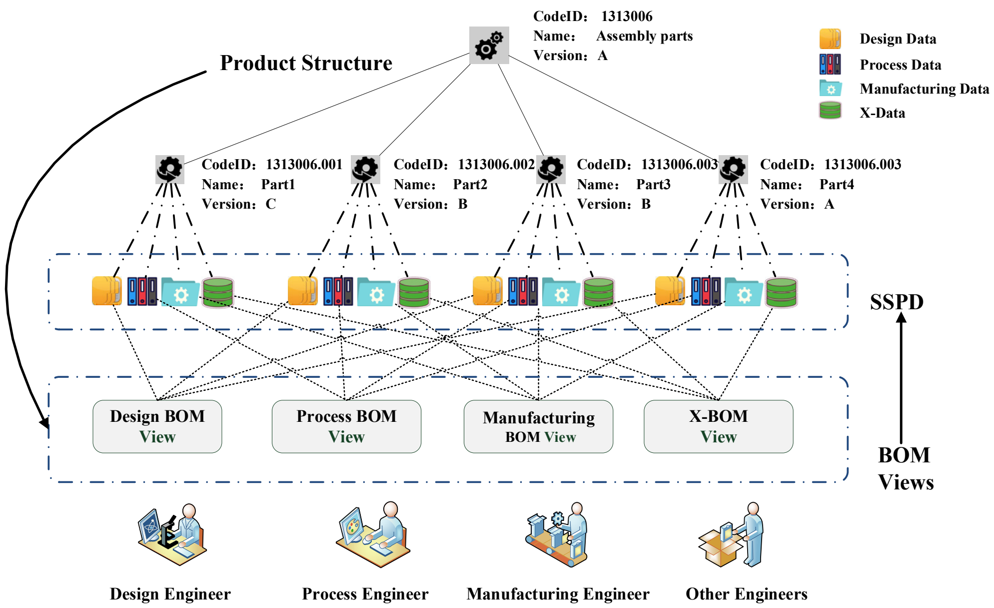

Figure 1. Model of BOM multi-view management for complex product.

In the framework, the product structure runs through the life cycle stages of design, process, and manufacturing and so on. Different stages can be expressed in different BOM forms. Figure 1 shows the three main types including design BOM, process BOM and manufacturing BOM as well as XBOM extended types (X can represent multiple business areas).

So as to facilitate the management and maintenance of various product data, the framework will provide a variety of corresponding BOM views, including design BOM view, process BOM view, manufacturing BOM view and XBOM view. As a result, various product data are aggregated, displayed and managed in the corresponding BOM view. For example, the design BOM view will be associated with design files, and the process BOM view will be associated with process specification files. Different types of engineers will centrally manage and maintain their business information through the corresponding BOM view.

The BOM multi-view can reflect the different expressions of the product structure at each stage of the product life cycle as well as there is a relationship between upstream and downstream, even if the product data of various stages and fields are scattered in different systems, through the BOM multi-view, the organization, management and maintenance of product information can be closely related to the product structure and form a logical single data source.

Furthermore, engineering changes are inevitable in the life cycle of a product as result of many causes including customer demand changes, design changes, process changes, equipment adjustments and error corrections. Each time a project is changed, a new version will be generated. Version changes will also have an important impact on the product structure. In order to efficiently maintain the relationship between them, this paper would integrate with product structure, BOM view and version to build a product structure model oriented to multi-view-version, as shown in Figure 2.

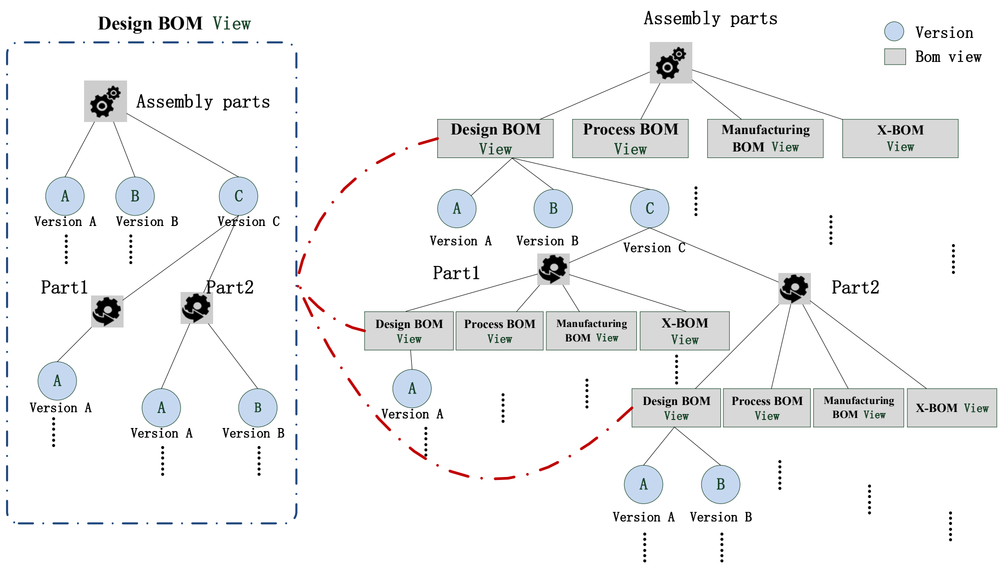

Figure 2. Product structure model oriented to multi-view-version.

This model contains three groups of elements: product structure, BOM view and version. Firstly, each BOM view contains a collection of BOM nodes of the product at the corresponding stage. Secondly, from top to bottom, the BOM node will be associated with owning versions, and then each version will be linked to the corresponding lower-level BOM node. Finally, through the iterative association between the BOM node and the version, the product structure with full version information is obviously presented through the BOM view.

4. The proposed methods of BOM multi-view management and version control for complex products

4.1. Product structure expression algorithm for multi-view-version BOM

In order to facilitate the work requirements of different engineers, it is necessary to filter out the precise product structure from system in term of rules. The specific process is shown in Figure 3.

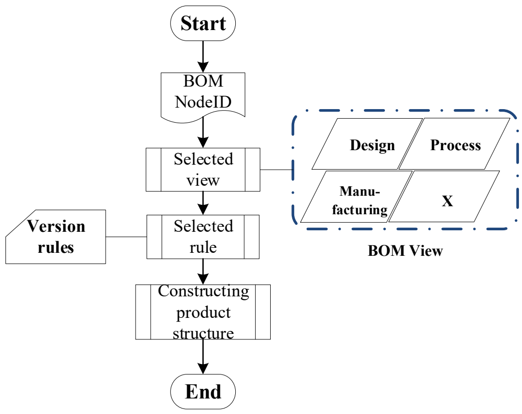

Figure 3. Process of product structure expression algorithm.

```
Function CreatProTree (Problem) returns a solution
Input: TreeNode_NodeHead, String_BomView, String_verRule and String_Version;
Output: TreeNode

If (NodeHead==Null)
    Return null;
Else
   Create Queue ← Queue<BomNode>BomQueue=new LinkedList< >;
   BomQueue ← GetChildrenBOMNodes(NodeHead, BomView, VerRule, Version);
   While (!BOMQueue.isEmpty())
       BomNode Child ← BOMQueue.poll( );
       TreeNode Node ← new TreeNode( );
       Node.setParentID(NodeHead.getID( ));
       Node.setID(Child.getID( ));
       Node.setName(Child.getName( ));
              ......
       NodeHead.add(Node);
       CreatProTree(Node, BomView, VerRule, Child.getVersion);
   End While
End IF
Return NodeHead;
```

Figure 4. Pseudo code of product structure expression algorithm for multi-view-version BOM.

Throughout the expression process, users can set conditions by business needs, including the specified view and version. Among them, the version limits the version requirements of the object (for example, select the latest version, valid version or a specific version); the view limits the view type of the product structure (design, process, manufacturing, etc.). Finally, the model obtains the BOM version nodes corresponding to the view from the system by recursive method, and forms precise structure tree. The pseudo code of the algorithm is shown in Figure 4.

4.2. Modeling of product BOM version control based on improved polychromatic graphs

4.2.1. Analysis of common BOM version control models

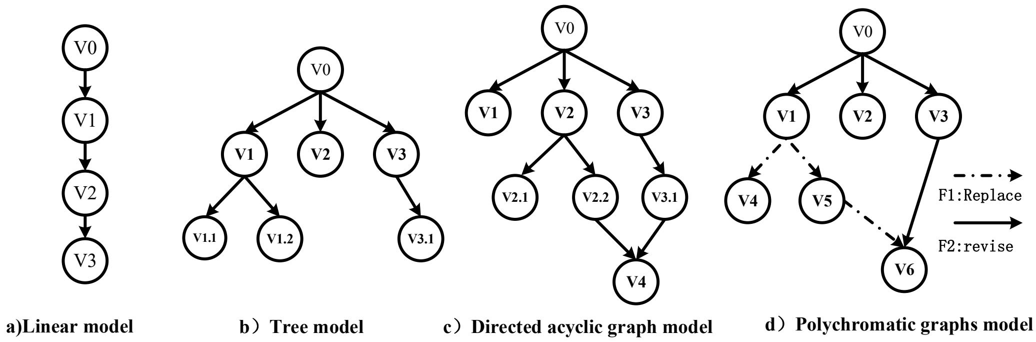

Figure 5. Traditional version management model.

Currently, the most widely used version management models include linear model, tree model, directed acyclic graph model and polychromatic graphs model, as shown in Figure 5.

In linear model, each version can only have a unique identifier; new version is automatically inserted at the end of the chain. This model can well describe the process of version sequence generation, but not easy to distinguish the difference between a replacement version and a revised version.

In tree model, a parent version can have multiple child versions. The model structure is clear, but it's difficult to represent the situation where multiple design versions are merged into a new version.

The version model of directed acyclic graph can describe the historical information of each version, and then it can also express the merging of multiple versions into a new version, but it's hard to express the relationship between the versions;

The polychromatic graphs model can represent the logical hierarchy of the version and relationship between a certain object version, such as the revision and replacement relationship, but it's less concerned about constraint relationships among different versions from distinct objects.

Therefore, this article will propose a BOM version control model based on the improved polychromatic graphs, which can be used to describe and trace BOM version evolution oriented to multi-object and multi-constraint relations in intelligent manufacturing process.

4.2.2. Introduction to polychromatic graphs theory

In the polychromatic theory [17] and its expanding application [18,19], polychromatic set

$A = (a_1, a_2, \cdots, a_n)$, a_i is defined F(A) as element to represent research object. $F(a_i)$ is corresponding to color of each element. $F(A) = (F_1, F_2, \cdots, F_m)$ expresses the color set of entireties of polychromatic set, in which F_j denotes the constitutes of $F(A)$.

Actually, the existence of unified color sets $F_j(A)$ is based on the existence of the individual color $F_j(a_i)$ which can be represented by a Boolean matrix $\|M_{i(j)}\|_{A,F(A)}$:

$$\begin{array}{cc} & \begin{array}{ccccc} F_1 & \cdots & F_j & \cdots & F_m \end{array} \\ [A \times F(A)] = & \begin{bmatrix} M_{1(1)} & \cdots & M_{1(j)} & \cdots & M_{1(m)} \\ \vdots & & \vdots & & \vdots \\ M_{i(1)} & \cdots & M_{i(j)} & \cdots & M_{i(m)} \\ \vdots & & \vdots & & \vdots \\ M_{n(1)} & \cdots & M_{n(j)} & \cdots & M_{n(m)} \end{bmatrix} \begin{array}{c} F_1 \\ \vdots \\ F_j \\ \vdots \\ F_m \end{array} \end{array} \tag{1}$$

Among them, the color F_j is expressed as a logical variable:

$$F_j = \begin{cases} 1, & F_j \in F(a_i) \\ 0, & F_j \notin F(a_i) \end{cases} \tag{2}$$

IF $F_j \in F(a_i)$ and $F(a) = \overset{i=1}{\underset{n}{\cup}} F(a_i)$, then $M_{i(j)} = 1$.

In addition, a Boolean matrix $[A \times F(A)]$ is used to denote the relationship between colors and constituent elements. The constituents of the elements of the entity which guarantees the existence of all unified colors of polychromatic sets can be represented by the following Boolean matrix as $[A \times A(F)]$, in which $A(F)$ is the composition of all the entities of unified colour. Thus, the polychromatic set can be expressed as a six-tuple:

$$PS = \{A, F(a), F(A), [A \times F(a)], [A \times F(A)], [A \times A(F)]\} \tag{3}$$

The polychromatic graphs are produced in the process of coloring the graphs, which can color nodes and edges by multiple colors. Usually, it consists of three components: $PG = (F(G), PS_A, PS_c)$, where $F(G)$ represents the overall coloring of the polychromatic graphs; PS_A represents the coloring of the nodes; PS_c represents the coloring of the edges:

$$PS_A = \{A, F(a), F(A), [A \times F(a)], [A \times F(A)], [A \times A(F)]\} \tag{4}$$

$$PS_C = \{A, F(c), F(C), [C \times F(c)], [C \times F(C)], [C \times A(C)]\} \tag{5}$$

When modeling with polychromatic theory, diverse colors would represent different attributes of objects. Besides, the distinct coloring of nodes and edges can also be used to reflect the relationship and constraint conditions between objects. Besides, some elements can be appropriately ignored or expanded during the modeling process to be adapted to the actual situation.

4.2.3. BOM version control model based on improved polychromatic graphs

During the process of R&D, manufacturing and maintenance of complex products, BOM needs to go through a large number of revisions and iterations and frequent version changes. Designers have to require frequent access to the previous version so as to inquire association relationship for analysis of irrationality and improvement plan. Meanwhile, the intermediate data in the work process also needs to be saved where there is also diversity in version status.

Aim to achieve effective management and precise traceability of the BOM version, this article combines advantages of traditional models and proposes a BOM version control model based on improved polychromatic graphs, as follows:

Definition 1: BOM version model of complex product based on polychromatic graphs

$$PGVM = (F(G), PS_N, PS_E, PR) \tag{6}$$

Where:

$F(G)$: Basic information of each object version space;

PS_N: Polychromatic collection of nodes in the object version spaces;

PS_E: Polychromatic collection of association relations in object version spaces;

PR: Constraint relations among object version spaces.

Definition 2: $F(G)$ mean basic information of each object version space such as object identifier, belongs, version number and the index of various data, etc.

Definition 3: Polychromatic collection of nodes in the object version spaces

$$PS_N = \big(A, A \times A, F(a), A \times F(a)\big) \tag{7}$$

Where:

A: Version collection of the object version spaces, corresponding to the node collection in polychromatic graphs;

$A \times A$: Version connection relationship in the object spaces and $A \times A = C$, iteration process of version can be traced;

$F(a) = \big(F_1(a), F_2(a), \cdots, F_n(a)\big)$: Version feature information of the object version spaces, including status and valid condition information, such as working version, released version, valid version, etc. It corresponds to the color set of the nodes of polychromatic graphs;

$A \times F(a)$: Coupling relationship between version and version feature in object version spaces, describing the version feature of each object.

Definition 4: Polychromatic collection of association relations in object version space

$$PS_E = \big(C, F(c), C \times F(C)\big) \tag{8}$$

Where：

$C = \big(c_{0(1)}, c_{0(2)}, \cdots, c_{j(k)}\big)$: Corresponding to set of colored edges in the polychromatic graphs, $c_{j(k)}$ represents the connection relationship between version nodes A_j and A_k;

$F(C) = \big(F_1(c), F_2(c), \cdots, F_n(c)\big)$: Characteristic information of relationship among versions of the object version spaces, such as revision, replacement, etc., represented by color set of the directed edge in the polychromatic graphs;

$C \times F(c)$: Coupling characteristic relationship between each version.

Definition 5: Constraints between each object version space

$$PR = (PR_1, PR_2, \cdots, PR_n) \tag{9}$$

Where:

PR_1: Timing;

PR_2: Mandatory;

PR_3: Optional;

PR_i: Self-defined.

If there are constraint relationships among the object version spaces, the associated version can be marked as $c_{i(j)}^{k(m)}$ by a colored directed edge, which means that there is a constraint relationship between version node *m* of the object *k* and the version node *j* of the object *i*.

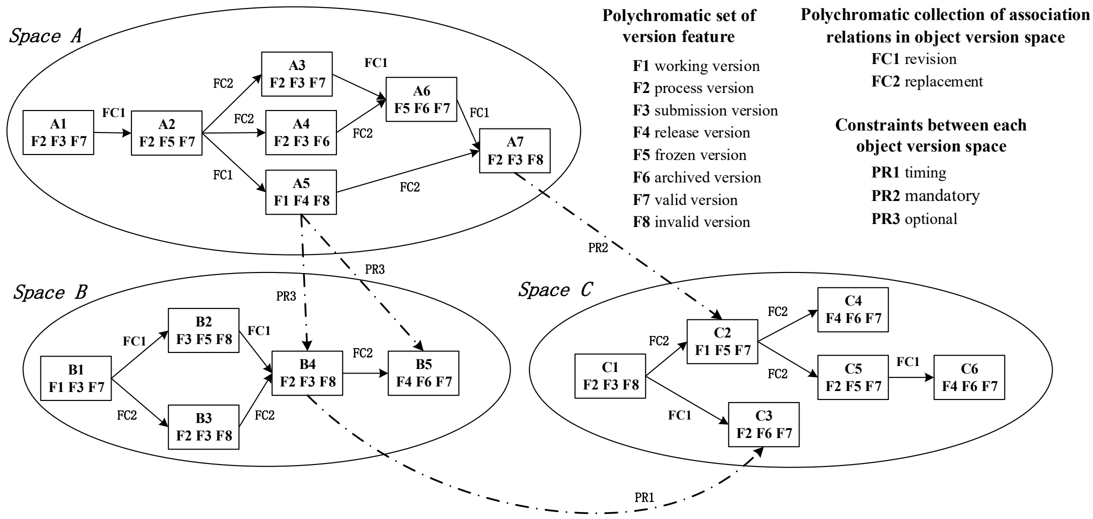

Figure 6. BOM version control model based on polychromatic graphs.

BOM version control model based on polychromatic graphs is shown as Figure 6. In the schematic diagram, the object version spaces ($F(G)$) contains Space A-C, and polychromatic set of version feature is represented by $F(a) = (F1, F2, \cdots, F8)$, specific meanings refer to F1–F8. The polychromatic set of node relations in the object version space is represented by $F(C) = (FC1, FC2)$, where $FC1$ denotes revision relationship, and $FC2$ denotes replacement relationship. The constraint relationship among various object version spaces is represented by $PR(C) = (PR1, PR2, \cdots, PRn)$, whose meanings have been marked in figure. To ensure the flexibility of model, the definition of the polychromatic set can be appropriately deleted or expanded to fit actual situation. The application principle of model will be explained in the following section.

5. Illustrate example

5.1. Application of product structure for multi-view-version BOM

With the development of the prototype system, product structure management function oriented to multi-view-version BOM is realized. The main interfaces are shown in Figures 7–9.

Referring to version rules, this function supports exhibiting, editing and changing the product structure in selected view. Meanwhile, it can display data information (such as documents, 2D/3D drawing and technological equipment) linked to the BOM node chosen as well as version collection, which is convenient for various engineers in corresponding BOM view to achieve related operations.

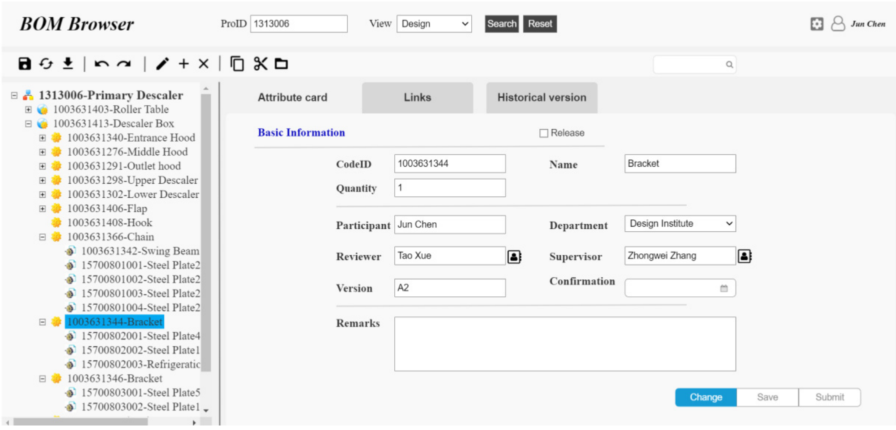

Figure 7. Application interfaces of product structure browser for multi-view-version BOM-Attribute Card.

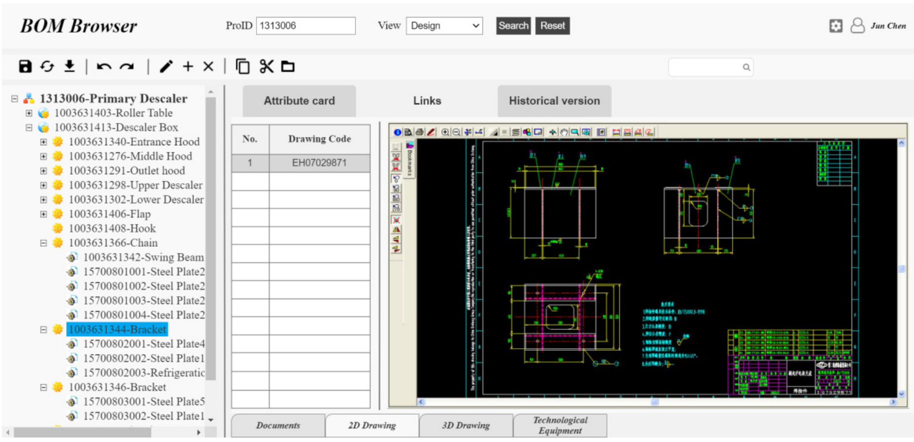

Figure 8. Application interfaces of product structure browser for multi-view-version BOM-Links.

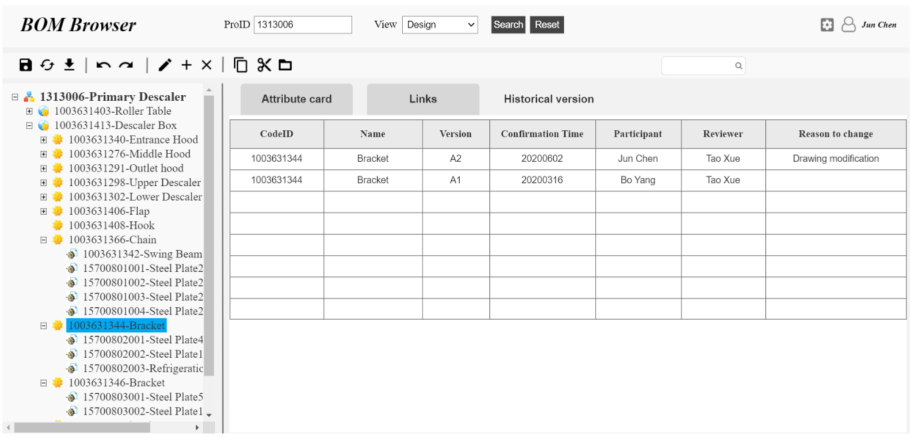

Figure 9. Application interfaces of product structure browser for multi-view-version BOM-Version Lists.

5.2. Application of BOM version control based on improved polychromatic graphs

Taking *Space A* in Figure 6 as an example, there is a set of object versions $A = (A_1, A_2, A_3, A_4, A_5, A_6, A_7)$. Given $A \times A$ Boolean matrix, it represents version iteration process in Space A as shown in Table 1, where $M_{ij} = 1$ means that A_i corresponding to row *i* is the parent version of A_j corresponding to the column *j*. For example, $M_{23} = M_{24} = M_{25} = 1$, then *A2* is the parent version of *A3*, *A4*, and *A5*. It's easy to trace the version iteration of object *A*.

Table 1. Version traceability information $A \times A$.

A	*A1*	*A2*	*A3*	*A4*	*A5*	*A6*	*A7*
A1	0	1	0	0	0	0	0
A2	0	0	1	1	1	0	0
A3	0	0	0	1	0	1	0
A4	0	0	0	0	0	1	0
A5	0	0	0	0	0	0	1
A6	0	0	0	0	0	0	1
A7	0	0	0	0	0	0	0

The features of each version in *Space A* can be obtained by using an $A \times F(a)$ polychromatic Boolean matrix, as shown in Table 2. For instance, $M_{65} = M_{66} = M_{67} = 1$ means that characteristics of version A_6 are *F5, F6* and *F7*, which are respectively frozen, archived, and valid.

Table 2. Version polychromatic information $A \times F(a)$.

A	*F1*	*F2*	*F3*	*F4*	*F5*	*F6*	*F7*	*F8*
A1	0	1	1	0	0	0	1	0
A2	0	1	0	0	1	0	1	0
A3	0	1	1	0	0	0	1	0
A4	0	1	1	0	0	1	0	0
A5	1	0	0	1	0	0	0	1
A6	0	0	0	0	1	1	1	0
A7	0	1	1	0	0	0	0	1

$A \times A = C = \left(c_{1(2)}, c_{2(3)}, c_{2(4)}, c_{2(5)}, c_{3(6)}, c_{4(6)}, c_{5(7)}, c_{6(7)}\right)$ is deduced from Table 1. Furthermore, polychromatic Boolean matrix $C \times F(c)$ will display association relationship of each version in *Space A*, as shown in Table 3. In terms of Table3, version A_1 and A_2 are revision relations, version A_5 and A_7 are replacement relationship and so on.

Owing to the situation that BOM structure of complex product is complicated and various parts have certain constraints at each stage of product life cycle, it's also necessary to express and deal with constraint relationships among different object version spaces. Thus, the polychromatic Boolean matrix $C \times PR(c)$ will be provided for Figure 6, as shown in Table 4.

Table 3. Association relationships of each version $C \times F(c)$.

C	$FC1$	$FC2$
$c_{1(2)}$	1	0
$c_{2(3)}$	0	1
$c_{2(4)}$	0	1
$c_{2(5)}$	1	0
$c_{3(6)}$	1	0
$c_{4(6)}$	0	1
$c_{5(7)}$	0	1
$c_{6(7)}$	1	0

Table 4. Constraint relationships among different object version spaces $C \times PR(c)$.

C	$PR1$	$PR2$	$PR3$
c_{B4}^{A5}	0	0	1
c_{B5}^{A5}	0	0	1
c_{C2}^{A7}	0	1	0
c_{C3}^{B4}	1	0	0

It can be obtained from matrix that there is an optional relationship between *A5* of *Space A* and *B4*, *B5* of *Space B*, that is, if object *A* selected version *A5*, object *B* would select either version *B4* or *B5* to meet requirement. And then, there is a mandatory relationship between *A7* of *Space A* and *C2* of *Space C*, that is, if object *A* selected version *A7*, object *C* would must select version *C2*. Moreover, there is a timing sequence relationship between *B4* of *Space B* and *C3* of Space *C*, that is, the version *C3* would be generated after completion of version *B4*.

5.3. Discussions

The multi-view-version BOM can make different expressions of the product structure at each stage of the product life cycle. With various views, the organization, management and maintenance of product information can be closely related to the product structure and form a logical single data source, which could ensure the consistency, completeness and accuracy of BOM information. Meanwhile, according to version rules, the product structure information in different view can be exhibited, edited and refreshed, which is convenient for various engineers in corresponding BOM view to achieve related operations collaboratively.

Combining with characteristics of common BOM version control models (section 4.2.1), the proposed version control method based on improved polychromatic graphs can better express version evolution process and represent relationship among versions. Furthermore, it can clearly reflect the hierarchical structure and merging process of revision during the version evolution process. Finally, the model can also accurately indicate the constraint relationship of versions in diverse object version spaces, such as timing, optional and mandatory, etc.

In summary, the model and method proposed can be adopted to quickly and accurately respond to product structure data, BOM version iteration information and constraint relationships among complex multiple objects so as to realize product data management and evolution tracking of BOM version efficiently. It’s beneficial to promote operation and optimization of various activities in the product life cycle.

6. Conclusions and future work

With widespread implementation of intelligent manufacturing and the growing complexity of modern product structures, the types and quantities of data in the product life cycle are increasing. Moreover, engineering changes in each stage of the life cycle are becoming more frequent. These make the management and maintenance of product data more complicated in the area of completeness, consistency, accuracy and traceability.

With the BOM multi-view management, not only can the product data scattered in various information systems be organized into a logically single data storage entity, and strict constraints and version association relationships can be established, thereby forming a single data source. These would better reply to SSPD requirements, and also realize effective association, management and maintenance of data at each stage of the product life cycle.

Moreover, the BOM version control method based on improved polychromatic graphs is adopted to quickly and accurately respond to BOM version iteration information, status and constraint relationships among complex multiple objects so as to realize accurate tracking of BOM version evolution.

The model and method proposed in paper can not only satisfy current product data management and maintenance of design, process and manufacturing, but also provide a flexible expansion strategy to dynamically define various types of BOM views and efficiently maintain related version information. In the following research, UML and Blockchain [20] would be combined with the model proposed, which could enhance the intelligibility and visualization of the model so as to possess a wide range of application.

Acknowledgments

This work was supported in part by the Key Laboratory of Expressway Construction Machinery of Shaanxi Province (Grant No. 300102250510), the Natural Science Basic Research Project of Shaanxi Province, China (Grant No. 2019JM-073), the China Postdoctoral Science Foundation (Grant No. 2018M633439) and Research Foundation of Xi’an Polytechnic University (Grant No. BS201847).

Conflict of interest

The authors declare no conflict of interest in this paper.

References

1. P. Burcher, Bill of materials, in *Wiley Encyclopedia of Management*, John Wiley & Sons, Ltd, **10** (2015), 1–1.
2. Z. Wei, H. Pan, X. Wang, Study on product BOM multi-view data definition based on object-oriented technique, *Period. Ocean Univ. China*, **50** (2004), 667–672.

3. R. Goncalves, D. Romero, A. Grilo, Factories of the future: challenges and leading innovations in intelligent manufacturing, *Int. J. Comput. Integr. Manuf.*, **30** (2017), 4–13.
4. X. Liu, X. Huang, Y. Ma, Q. Meng, Y. Meng, Research on xBOM for product whole life cycle, *Comput. Integr. Manuf. Syst.*, **8** (2002), 983–987.
5. A. O. Aydin, A. Güngör, Effective relational database approach to represent bills-of-materials, *Int. J. Prod. Res.*, **43** (2005), 1143–1170.
6. X. Huang, Y. Fan, Research on BOM views and BOM view mapping model, *Chin. J. Mech. Eng.*, **41** (2005), 101–106.
7. F. Xiang, Y. Huang, Z. Zhang, G. Jiang, Y. Zuo, Research on ECBOM modeling and energy consumption evaluation based on BOM multi-view transformation, *J. Ambient Intell. Hum. Comput.*, **10** (2019), 953–967.
8. C. Zhou, X. Liu, F. Xue, H. Bo, K. Li, Research on static service BOM transformation for complex products, *Adv. Eng. Inf.*, **36** (2018), 146–162.
9. S. Cai, D. Hu, Y. Fei, J. Xu, Research on BOM multi-view mapping method orient to aircraft in-service data management, *J. Phys.: Conf. Ser.*, **1550** (2020), 032124.
10. N. Xia, J. Hu, W. Wang, Managing and mapping method for BOM multiple-view based on transform-table, *J. Shanghai Jiaotong Univ.*, **42** (2008), 584–589.
11. T. Li, Y. Li, W. Sun, G. Zhao, Research and implementation of version management method for product structure information tree, *Appl. Res. Comput.*, **27** (2010), 3829–3833.
12. H. Zhang, S. Shi, Q. Shi, PLM-oriented BOM data management and practice, *Manuf. Autom.*, **40** (2018), 30–34.
13. F. Liu, Y. Wang, Optimization of multi-version distribution mode for directed acyclic graph structure, *Comput. Eng.*, **37** (2011), 74–76.
14. Y. Chen, Y. Liu, J. Liu, F. He, Design of PDM multi-version management model based on polychromatic sets theory, *Advanced Materials Research*, **271** (2011), 28–33.
15. L. He, X. Ming, F. Kong, H. L. Zuo, Z. H. Yao, Research on airplane BOM management based on single source of product data, *Appl. Mech. Mater.*, **232** (2012), 184–188.
16. M. Abam, S. Har-Peled, New constructions of SSPDs and their applications, *Comput. Geom.*, **45** (2012), 200–214.
17. V. Pavlov, *Polychromatic Sets and Graphs for CALS*, Moscow, Russia: STANKIN Press, 2002.
18. X. Gao, L. Xu, X. Wang, Y. Li, M. Yang, Y. Liu, Workflow process modelling and resource allocation based on polychromatic sets theory, *Enterp. Inf. Syst.*, **7** (2013), 198–226.
19. S. Lin, Y. Wang, L. Jia, System reliability assessment based on failure propagation processes, *Complexity*, **2018** (2018), 1–19.
20. J. Leng, G. Ruan, P. Jiang, K. Xu, Q. Liu, X. Zhou, et al., Blockchain-empowered sustainable manufacturing and product lifecycle management in industry 4.0: A survey, *Renewable Sustainable Energy Rev.*, **132** (2020), 110112.