Contents lists available at GrowingScience

International Journal of Industrial Engineering Computations

homepage: www.GrowingScience.com/ijiec

A dual-layer BOM change control model for efficiency improvement in ETO manufacturing

Chunhua Wan[a], Yufei Zeng[a], Ji Ma[b], Tao Wang[a*] and Kaiyang Zhong[c*]

[a]*School of Management Science and Engineering, Southwestern University of Finance and Economics, Chengdu, China*
[b]*School of Network Security, Jinling Institute of Technology, Nanjing, China*
[c]*College of Information Science & Electronic Engineering, Zhejiang University, Hangzhou, China*

C H R O N I C L E

Article history:
Received January 15 2025
Received in Revised Format March 16 2025
Accepted April 17 2025
Available online April 17 2025

Keywords:
ETO (Engineer-to-Order)
BOM (Bill of Materials)
Change control
Dual-layer traceability
Supply chain collaboration

A B S T R A C T

To address the frequent changes, dynamic evolution, and complex collaboration of BOM (Bill of Materials) under ETO (Engineer-to-Order) mode, this paper proposes a dual-layer BOM-based change control model. First, to enable model definition and change expression throughout the product lifecycle, a version control-based BOM model is defined by introducing material revision, material relationship links, and a multi-view mechanism, while also constructing a general BOM structure system. Then, to ensure traceability of product structural changes and cross-view consistency in the ETO mode, we design a dual-layer change traceability model. This model features vertical version chains and horizontal view collaboration traceability as its core components. Finally, an ETO-oriented BOM change operation model is constructed to standardize both in-view change operations and cross-view cooperative operations. This standardization enhances change control capability and lifecycle traceability efficiency of product structures in ETO manufacturing environments. The application of this model in a large equipment manufacturing enterprise shows that it significantly improves the change response efficiency and provides strong support for the digital transformation and supply chain collaboration of ETO enterprises.

1. Introduction

The global manufacturing industry operates across a diverse spectrum of production modes, including MTS (Make-to-Stock), MTO (Make-to-Order), ATO (Assemble-to-Order), and ETO (Engineer-to-Order). Different production modes have their own characteristics in terms of response to customer demand, production efficiency, and complexity (Gosling, & Naim, 2009; Kalantari et al., 2011). MTS is forecast-based mass production, which is suitable for products with stable demand and a high level of standardization. Its advantage lies in fast delivery, but the inventory cost is high and it is difficult to respond to demand fluctuations. MTO starts production after receiving customer orders, which can reduce inventory pressure, but suffers from longer production cycles and limited responsiveness to urgent demands. ATO takes into account inventory management and delivery speed by pre-producing standardized parts and completing assembly quickly after accepting orders, but it is limited in flexibility for highly customized requirements. As product complexity has increased over time, particularly in industries requiring high levels of customization and engineering expertise, ETO has emerged as a production approach well-suited to address these sophisticated manufacturing challenges (Pracucci, 2024; Thajudeen et al., 2022). ETO is suitable for high-complexity and high-value-added products, such as large-scale equipment manufacturing, aerospace, etc. (Mello et al., 2015; Love et al., 2024; Willner et al., 2016).

Among the above models, the ETO model has become increasingly important in high-end manufacturing because of its ability to maximize the satisfaction of individual customer needs (Jing, 2022). However, along with the high degree of customization comes significant complexity. In the ETO model, design and manufacturing are highly integrated. This integration involves

* Corresponding author
E-mail ccnuwt@swufe.edu.cn (T. Wang) zhongky@zju.edu.cn (K. Zhong)
ISSN 1923-2934 (Online) - ISSN 1923-2926 (Print)
2025 Growing Science Ltd.
doi: 10.5267/j.ijiec.2025.4.007

extensive outsourcing of design, production, and parts procurement, which creates substantial demands for collaborative supply chain management (Hicks et al., 2000). In addition, frequent changes in customer requirements require continuous optimization of design solutions (Vaagen et al., 2017), which not only increase the technical and cost risks, but also bring uncertainty to the delivery cycle (Bhalla et al., 2023).

In this context, BOM (Bill of Materials) serves as a core tool for manufacturing enterprises. It spans the entire processes of product development, manufacturing, and after-sales maintenance, becoming a key support for efficient enterprise operations (Qiao et al., 2023; Schramm et al., 2023). Various departments utilize BOM for different purposes. The design department plans part structures and relationships. The process department formulates manufacturing and assembly processes. The production department conducts actual production; the purchasing department ensures resource supply. The quality department maintains product consistency. The maintenance department provides after-sales service support. Especially in ETO mode, due to complete product customization, BOM exhibits a high degree of dynamism and requires frequent updates to manage multiple parallel versions, facilitate cross-departmental collaboration, and navigate complex approval processes.

Therefore, how to efficiently manage the BOM in ETO mode, coordinate the flow of information among departments, and reduce the cost and risk brought by changes is one of the key challenges for enterprises to achieve high-end manufacturing. At present, scholars' research in the field of BOM management mainly focuses on the reconstruction methods of BOM data and the conversion mechanisms of BOM view. In terms of BOM data reconstruction, researchers have proposed a variety of methods to optimize the data structure and organization of BOM. For example, Wang et al. (2022) developed a consistent reconstruction method of BOM based on digital twin technology for complex products to ensure the accuracy and consistency of BOM data. Fa et al. (2020) introduced a dynamic EBOM construction technique based on MBD (Model-Based Definition) model to quickly and accurately generate BOM structures from 3D design models. Chowdhury and Moon (2023) created a BOM-generating AI model (BOM-GAIM) based on Mask R-CNN and image segmentation techniques, which can automatically generate BOM. Wang et al. (2024) established a knowledge-driven cross-functional BOM (XBOM) reconstruction method for complex products to shorten the BOM reconstruction cycle and improve efficiency and quality. Shiau (2020) designed a decision support module based on data mining to generate planning BOM. In terms of BOM view conversion, scholars have devoted themselves to mapping and converting BOM data across different phases and departments. For example, Zhao et al. (2024) proposed a multi-view mapping method based on BOM, including the definition of material item description and material item relationship description. The authors constructed a BOM multi-view model for complex products and designed the operation rules and process of XBOM multi-view mapping. Zhou et al. (2018) developed a conversion method from EBOM to service BOM (SBOM) by constructing a composite structure of generic service BOM (G-SBOM) and instance service BOM (I-SBOM). This approach realizes static BOM data conversion and integrated management of complex products from the design and manufacturing stage to the maintenance and repair stage. Wang et al. (2023) established a BOM model conversion method for hierarchical production planning management of complex products; they constructed hierarchical BOM (HBOM) structures by integrating generic HBOM (G-HBOM), Pre-manufacturing Instance HBOM (I-HBOM(BM)), and Post-manufacturing Instance HBOM (I-HBOM(AM)) to realize the dynamic conversion and integrated management of BOM data in different planning management stages.

However, existing research has paid insufficient attention to the dynamic tracking and feedback mechanism of BOM data. Especially in the ETO mode, product design and manufacturing changes occur frequently. How to capture these changes in real time and effectively report them to all aspects of the product life cycle remains an urgent problem. The tracking and feedback of BOM data can not only provide support for design optimization, but also help enterprises promptly identify and solve problems in the production process. This is significant for improving product quality and production efficiency. Therefore, based on in-depth analysis of the characteristics of ETO production mode and the inadequacies of existing research, this study addresses the core problems in BOM change management under the ETO model. The main innovations include: (1) Redefining the BOM model based on version control, constructing a common BOM structure system by introducing material revisions, material relationship links, and a multi-view mechanism, and enabling the model to express changes throughout the product life cycle. (2) Proposing a dual-layer change traceability model that ensures change traceability and cross-view consistency of product structure in the ETO mode through a vertical version chain and a horizontal view collaborative traceability approach. (3) Establishing an ETO-oriented BOM change operation model to achieve efficient change management and full life cycle traceability of product structure in the ETO mode by standardizing intra-view BOM change operations and cross-view collaborative operations.

The structure of this paper is organized as follows: Section I introduces the variability and complexity of complex product manufacturing in ETO mode and analyzes the research background of BOM in response to the complexity of ETO. In Section II, we analyze the ETO business scenario and detail the BOM characteristics and change processes in the business operations of ETO. In Section III, we use the BOM change management model to define the BOM model and the change control model, and construct the ETO-oriented BOM operation model. In section IV, we discuss the application mode and effect of the model in the ETO process in a large enterprise. Finally, in Section V, we review the work of this paper and discuss future research directions.

2. ETO Business Scenario Analysis

2.1 ETO business process

Under the ETO manufacturing mode, the business process of a product from order taking to delivery exhibits distinctive characteristics. As shown in Fig. 1, unlike the traditional manufacturing mode, the business process under the ETO mode is primarily driven by engineering design as its core component. Each stage—from customer requirement analysis through conceptual design to detailed design—may involve multiple rounds of iteration and optimization (Lee et al., 2012). In a typical ETO project, the design phase accounts for about 30-40% of the entire project cycle. To shorten the project cycle, design, procurement, and manufacturing processes are often executed concurrently through parallel engineering approaches, which significantly increase the complexity of collaborative management. Numerous real-time decisions must be made during project execution, including design adjustments, material substitutions, and process optimizations.

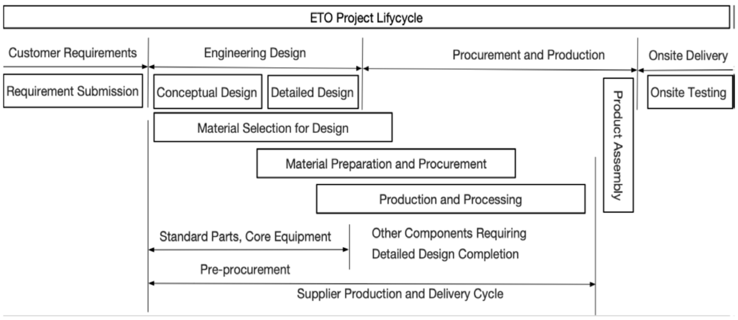

Fig. 1. ETO Model Business Processes

2.2 BOM characteristics in ETO mode

BOM in ETO mode is significantly complex and dynamic, which is reflected mainly in its evolution process and management characteristics. Regarding the evolution process, as shown in Fig. 2, BOM undergoes a complete transformation chain. This begins with requirement structure and functional structure, progresses to conceptual BOM (CBOM), then to entity structure, engineering BOM (EBOM), process BOM (PBOM), and finally forms Manufacturing BOM (MBOM). In this process, MBOM requires further refinement into various branches to meet the needs of different production organizations. These branches include outsourced production MBOM, self-manufactured parts production MBOM, procurement BOM, and service BOM (SBOM).

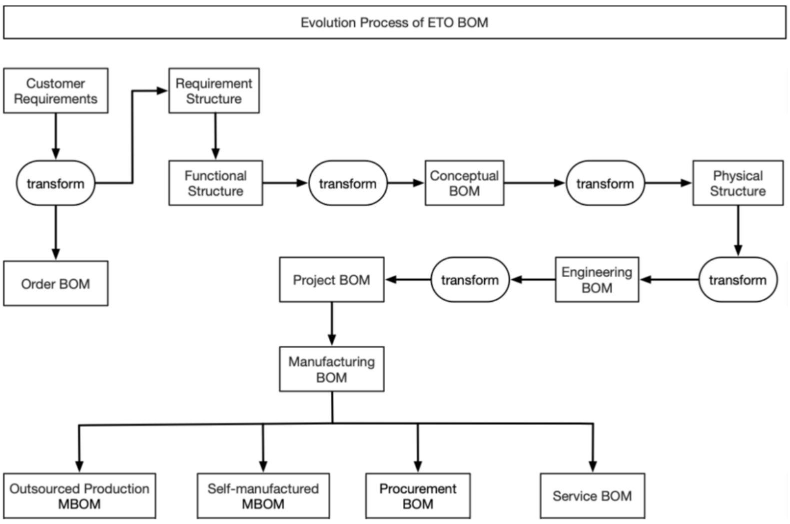

Fig. 2. the Evolution Process of BOM in ETO

Regarding management characteristics, the BOM under the ETO model presents two outstanding features: multiple releases and frequent changes, as shown in Fig. 3. The feature of multiple releases comes from the progressive nature of engineering design. Engineering design departments usually prioritize the release of BOM for standardized parts and long-cycle materials, along with their drawings, to reduce the overall manufacturing time. Although this strategy helps improve project efficiency, it also increases the complexity of BOM management. Frequent changes are the inevitable result of the design deepening process. These changes involve drawing modifications, material replacement, quantity adjustment, and other aspects, sometimes even occurring during component processing.

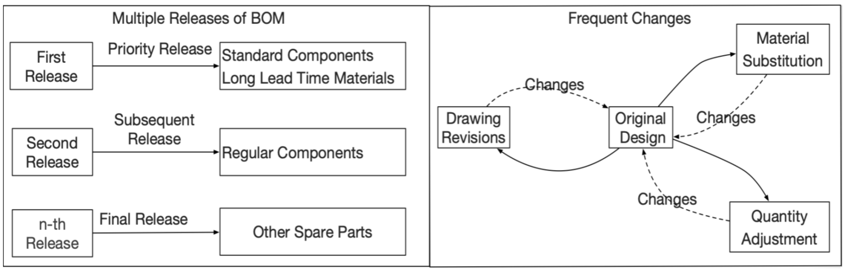

Fig. 3. BOM Management Features

2.3 Typical Change Scenario Analysis

In the ETO mode, BOM change scenarios are characterized by diversity and complexity. From the perspective of customer requirements, changes in functional requirements often trigger extensive BOM changes. Since ETO products are usually large, complex customized equipment or systems, their requirement changes often involve core engineering features. Typical changes include process parameter adjustments, changes in equipment operating environment, and changes in system integration requirements. For example, in the customization of large port machinery, the customer may require additional typhoon-proof structural design due to the terminal's actual situation. This necessitates comprehensive improvements to the equipment's foundation fixing methods and structural strength. These changes often trigger systematic adjustments to the design, which in turn leads to BOM changes at multiple levels, such as material specifications, component sizes, and part types. Compared with conventional products, changes in customer requirements under the ETO mode have characteristics of high technical complexity, extensive impact scope, and strong engineering relevance. Consequently, coordinated evaluation and processing by multiple departments—including design, process, and procurement—are often required.

From the design perspective, changes primarily stem from design optimization requirements. In ETO projects, due to the uniqueness and complexity of the product, the design process often needs to be continuously optimized and adjusted. Such changes mainly include structural adjustments for performance enhancement, adaptations to cope with field installation conditions, and local optimization due to design conflicts. For example, in large equipment design, optimizing key component structures to improve efficiency has widespread implications. Such optimization not only involves changes in component parameters themselves, but also requires corresponding adjustments to material selection, manufacturing processes, and assembly procedures. Additionally, material optimization represents another significant source of change. This may result from demands for improved material performance, cost-control driven substitutions, or adjustments necessitated by supply chain factors.

In the manufacturing process, process adaptation changes for ETO products are particularly critical. Such changes mainly include optimization of manufacturing processes, reduction of assembly complexity, and enhancement of processing capability alignment. Additionally, on-site feedback frequently triggers BOM changes. These changes typically address assembly process issues, incorporate trial operation feedback, and resolve quality problems. For example, process conflicts found during equipment assembly may require adjusting part of the structural design to optimize the assembly path. These on-site feedback-driven changes often require urgent attention, necessitating close collaboration among production, process engineering, design, and other departments to ensure rapid response and effective implementation.

Supply chain factors are particularly prominent in the ETO mode, mainly reflected in four dimensions. The first dimension concerns material availability, encompassing raw material supply shortages, substitution of discontinued standard parts, and delivery conflicts for long-cycle components. Addressing these issues typically requires identifying alternatives while ensuring compliance with technical requirements. The second dimension relates to supplier capacity matching. This involves supplier process capability assessment, quality assurance evaluation, and capacity load balancing. These supplier capability factors directly influence the selection of technical solutions for purchased and outsourced components. The third dimension addresses cost control demands, including measures to manage material price fluctuations, optimize purchasing strategies, and consolidate batch orders. During such cost-driven changes, stakeholders must establish a balance between technical feasibility and economic considerations. The fourth dimension concerns supply chain localization requirements. This primarily involves considerations of import substitution, local supplier integration, and after-sales support capabilities. Such localization efforts frequently necessitate systematic adjustments to technical programs. Changes driven by these supply chain factors exhibit significant uncertainty and broad organizational impact, necessitating collaborative assessment and decision-making across multiple departments, including procurement, technical, and quality control units.

Analysis of the previously discussed change scenarios reveals that BOM changes in the ETO mode exhibit four key characteristics: multiple sources, high correlation, critical timeliness, and significant complexity. Change sources span multiple domains including design, customer requirements, production processes, and supply chain management. Significantly, a modification in one domain frequently triggers chain reactions that impact interconnected processes throughout the project ecosystem. Changes may emerge at any stage of project execution, necessitating rapid response capabilities and agile management systems to minimize disruption to project timelines. Furthermore, effective change management demands cross-functional collaboration. It also necessitates careful balancing of technological feasibility, cost implications, and schedule constraints to achieve optimal project outcomes.

3. BOM change management model

3.1 BOM version definition model

The BOM version definition model is constructed upon four core elements: materials, material revisions, material relationship links, and structure views. These integrated elements collectively constitute a comprehensive product structure management system. Materials serve as the fundamental component of the product structure and can represent a specific part, a complex component, or a complete product system. To manage the continuous optimization and iteration of products throughout various stages including design, production, and implementation, the model incorporates the concept of material revision. Each material can have multiple revisions, with each revision documenting the technical characteristics, process parameters, and other information of the material at a specific stage. This versioning mechanism establishes comprehensive historical traceability, enabling stakeholders to track and analyze the complete evolutionary path of materials throughout the product lifecycle.

Material relationship links serve as the formal mechanism for expressing organizational hierarchies and associations between materials within the product structure. These relationship links possess explicit directionality, precisely defining hierarchical subordination and assembly relationships among interconnected materials. The BOM version definition model implements a flexible many-to-many association mechanism, wherein a single link can simultaneously accommodate multiple input material revisions and multiple output material revisions, thereby supporting complex product structures. Significantly, these relationship links are established directly at the material revision level rather than at the material itself, enabling precise version-specific connections. This revision-specific linkage architecture ensures exceptional precision and comprehensive traceability within the product structure, establishing a robust foundation for effective change management, configuration control, and lifecycle governance.

To address the diverse structural requirements across various operational scenarios, the BOM version definition model introduces the concept of BOM view. Each BOM view, comprising a specific collection of material revisions and their linking relationships, represents the product structure from distinct operational perspectives. For example, during product development, the engineering view prioritizes functional characteristics and technical parameters of components, whereas the manufacturing view emphasizes production processes, assembly sequences, and tooling requirements. This multi-view mechanism enables diverse organizational departments to access and manipulate precisely tailored structural information while maintaining consistency through a unified underlying product model.

Through this comprehensive hierarchical BOM version definition model, the product structure management system effectively captures and manages the evolution of product structures while simultaneously accommodating differentiated information requirements across various lifecycle stages and organizational roles.

3.1.1 Definition of material system

In product structure management, the material system is the most fundamental conceptual framework. The formal model begins with the definition of material collection M, which encompasses all base materials within the system:

$$M = \{m_1, m_2, \dots, m_n\}$$

where m_i denotes the i-th material. Each material m_i can have multiple revisions $m_{i\bullet j}$, which denotes the j-th revision of the i-th material, forming the revision set of the material:

$$V_i = \left\{m_{i\bullet 1}, m_{i\bullet 2}, \dots, m_{i\bullet j}\right\}$$

All revisions of all materials in the system constitute the complete set of material revisions V, which is the union of all material revision sets:

$$V = \bigcup_{i=1}^{n} V_i$$

To effectively manage the state of material usage, state machines are defined for each material revision to track its progression through the product lifecycle:

$$State(m_{i\bullet j}) \in \{Draft, Release, Obsolete\}$$

The Draft status indicates that the material version is in the process of being designed or modified. The Release status indicates that the material version has been confirmed ready for production. The Obsolete status indicates that the material version is no longer in use.

To accurately track the temporal validity of each material revision, two critical time attributes are defined: the effective time $t_{start}(m_{i\bullet j})$ and the expiration time $t_{end}(m_{i\bullet j})$. These timestamps establish the temporal boundaries for each material revision's active lifecycle, enabling precise product configuration management and comprehensive traceability analysis.

To ensure traceability of material changes, the description of differences $\Delta(m_{i\bullet j}, m_{i\bullet(j-1)})$ is recorded, detailing changes between adjacent versions. Additionally, the classification of change causes is documented, including design optimization, process improvement, and rectification of quality problems. These records serve as a foundation for subsequent change analysis and for extracting lessons learned from the material evolution process.

To ensure the consistency and reliability of the material system, two important global constraints must be satisfied. The first is the uniqueness constraint: $\forall m_{i\bullet j}, m_{k\bullet l} \in V, (i,j) \neq (k,l) \Rightarrow m_{i\bullet j} \neq m_{k\bullet l}$, which ensures that each material revision is unique in the system. The second is the revision partial order relationship: $m_{i\bullet j} < m_{i\bullet(j+1)}$, which ensures a linear evolution of revisions, forming a clear developmental chain and facilitating the tracking of material development history.

Fig. 4 illustrates the complete state migration process for material revisions. The process begins with the "Draft" state, which consists of three sub-stages: from "New", through "Design" to "Evaluation". In the Evaluation sub-stage, if material revisions require changes, they return to the Design sub-stage through the "revise" path for iteration. If the evaluation passes, the material revision proceeds to the "Release" state via the "approve" path. If the evaluation fails, the material revision enters the "Obsolete" state through the "reject" path. When the released material revisions complete their lifecycles, they enter the "Obsolete" state via the "achieve" path and eventually reach the deprecation point. This state migration diagram clearly describes the full lifecycle management of material revisions from creation to deprecation.

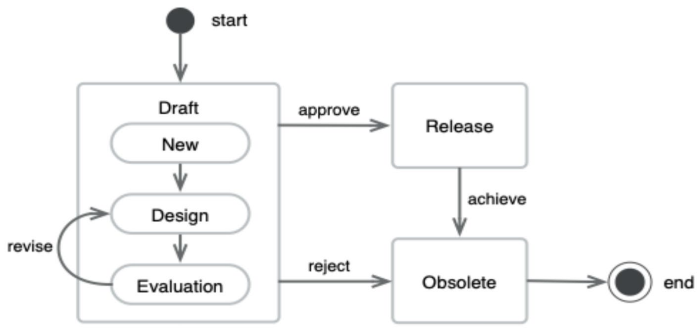

Fig. 4. Material Revision Status Migration Diagram

3.1.2 Material Relationship Link Definition

The Material Relationship Links framework systematically defines the various relationships and interactions between materials. This is accomplished through the link set R, which contains all possible material relationship links. This set can be represented as:

$$R = \{r_1, r_2, \dots, r_n\}$$

In this system, each specific link r_i is defined as a triad:

$$r_i = (I_i, O_i, t_i)$$

Within the triad, the input item $I_i \in V \times \mathbb{N}^+$ defines the revision of the input material and its specific usage in the form of ordered pairs. For example, $(m_{1\bullet 2}, 3)$ means that the 2nd revision of material m_1 is required and 3 units are needed. This representation not only specifies the material version, but also quantifies the usage requirement. The output item $O_i \in V \times \mathbb{N}^+$, on the other hand, specifies the revision of the output material and its output quantity. This is also represented in the form of ordered pairs, clearly defining the material and its quantity after production or conversion. The link semantic type t_i indicates the nature of the relationship between materials. This includes, but is not limited to: Assembly (assembly relationship, indicating that multiple materials are assembled to form a new material), Dependency (dependency relationship, indicating necessary dependencies among materials), Substitute (substitution relationship, indicating that materials can be substituted for each other), and so on. This categorization helps the system to understand and deal with different types of material relationships.

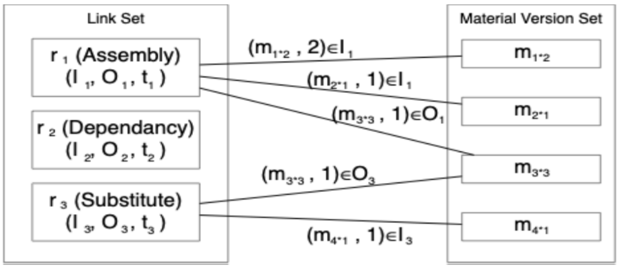

Fig. 5. Material Relationship Linking Model Diagram

Fig. 5 illustrates the linking model of relationships among material versions, which consists of two core components: the Link Set on the left and the Material Version Set on the right. Three basic types of relationships are defined in the Link Set: r_1 denotes Assembly, r_2 denotes Dependency, and r_3 denotes Substitute, each containing an input set (I), an output set (O), and a temporal attribute (t). These relations are linked to the material versions on the right (e.g., $m_{1\bullet 2}$, $m_{2\bullet 1}$, etc.) via link expressions in the center. The link expression is described as $(m_{i\bullet k}, n) \in I_k \vee O_k$, which denotes the mapping relationship between a certain material version and a specific relation type. Here, n represents the quantity of the material version used or produced in the relationship. This model structure can clearly describe the assembly, dependency, and substitution relationships among different material versions, and provides a formalized expression for material management.

3.1.3 BOM view structure definition

In an ETO-oriented BOM management system, views are regarded as multi-dimensional product structure expression carriers, and their hierarchical definition and version control mechanisms are the key to supporting complex change management. We adopt a multi-view collaborative architecture to address these challenges. This approach realizes dynamic evolution and consistency maintenance of BOM data under different business scenarios through abstraction modeling of view collections.

First, the system supports multiple logically independent BOM views to form a view set:

$$W = \{w_1, w_2, \dots, w_n\}$$

Here, each view $w_i \in W$ represents a product structure representation (e.g. design view, process view, purchasing view) of a specific dimension. Data linkage among views is achieved through cross-view dependency rules. For example, design changes need to be synchronized to the manufacturing view.

Each view w_i contains a strictly chronological sequence of commit versions:

$$S_i = \{s_{i\bullet 1}, s_{i\bullet 2}, \dots, s_{i\bullet j}\}$$

where $s_{i\bullet j}$ denotes the j-th commit version of the view w_i, and its data structure is defined as:

$$s_{i\bullet j} = (V_{i\bullet j}, R_{i\bullet j})$$

where $V_{i\bullet j}$ is the set of material revisions included in the commit, and $R_{i\bullet j}$ is the set of links included in the commit. This structure helps to understand and track the relationships between the components of the product and how they have changed over time.

By aggregating all the commit versions in each view, we can get a set consisting of all the commit versions of all the views in the whole system:

$$S = \bigcup_{i=1}^{n} s_i$$

Fig. 6 depicts the hierarchical versioning management mechanism of the BOM version control system. The system manages the BOM structure in different scenarios through a multi-view mechanism. All views constitute a BOM view set $W = \{w_1, w_2, \dots, w_n\}$, and each view represents a specific BOM usage scenario or perspective. Under each view (e.g., w_1), the change history of the BOM is recorded through a commit set (Commits) $S_i = \{s_{i\bullet 1}, s_{i\bullet 2}, \dots, s_{i\bullet j}\}$. These commits record all the modifications of the BOM in this view in chronological order. Each commit (e.g., $s_{i\bullet j}$) contains two core elements: a material version set ($V_{i\bullet j}$) that records the material version information at the moment of the commit, and a link set ($R_{i\bullet j}$) that defines the correlation among these material versions. This hierarchical structure enables parallel management of multiple views, version history tracking, and material relationship management, allowing the system to flexibly manage the BOM structure in different scenarios while maintaining complete change history and relationship tracking.

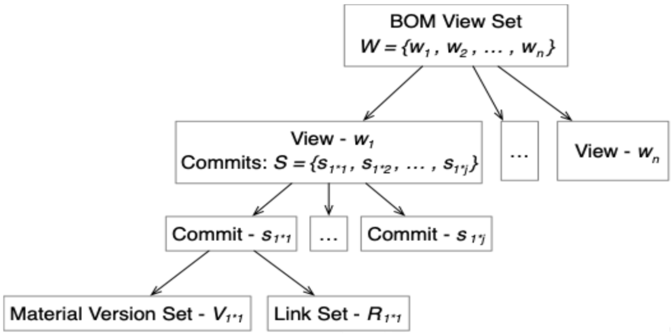

Fig. 6. Structure of view-based version control system

3.1.4 Model constraints

In the BOM change management model, a series of constraints are defined to ensure data accuracy and consistency:

(1) Acyclicity constraint: the graph $G = (V, E)$ corresponding to the commit revision in each view must be a Directed Acyclic Graph (DAG). This means that, to avoid potential data conflicts and logical errors, no circular dependencies are allowed in the dependency relationships between materials and their revisions, where $V = V_i$, the set of edges $E = \{(x, y) | \exists r \in R_i, x \in r.I \wedge y \in r.O\}$, where $r.I$ and $r.O$ denote the input and output endpoints of the link r, respectively. This constraint ensures that the data flow and material dependencies are clear and reasonable, which helps to simplify data analysis and problem tracking.

(2) Referential Integrity: $\forall r \in R, I \cup O \subseteq V$. This constraint requires that all links can only refer to material revisions that already exist, that is, for any link $r \in R$, its set of input and output endpoints $I \cup O$ must be a subset of the set of nodes V. This means that any attempt to create a link pointing to a material revision that does not exist within this system will be blocked, thus preventing dangling references and inconsistent data states. The referential integrity constraint ensures database integrity and data consistency, allowing the system to reliably reflect the actual composition of the product.

(3) Temporal validity: this constraint ensures that the time ranges of all links are compatible with the time ranges of the material revisions they reference. Specifically, for any link $r \in R$, its end time $t_{end}(r)$ must not be earlier than the maximum of the start times of all material revisions it references, and must not be later than the minimum of the end times of those material revisions. It is formally expressed as $\forall r \in R, max_{v \in I \cup O} t_{start}(v) \leq t_{end}(r) \leq min_{v \in I \cup O} t_{end}(v)$. This constraint effectively avoids data clutter caused by time mismatches, and ensures that any changes made to a product during its lifecycle are based on the most current and valid information available.

3.2 BOM Change Traceability Model

Under the ETO mode, frequent changes in product structure impose strict traceability requirements on BOM management. This section proposes a dual-layer traceability mechanism to achieve full-lifecycle management of product structure changes.

3.2.1 Layered model of traceability relationship

In order to build a full-lifecycle change tracing capability, this model designs a dual-layer tracing mechanism: change tracing and collaborative tracing. Change tracing reflects the vertical update dependency within the same view, constituting a strict version chain; collaborative tracing reflects the conversion association across views, forming a horizontal collaborative evolution relationship.

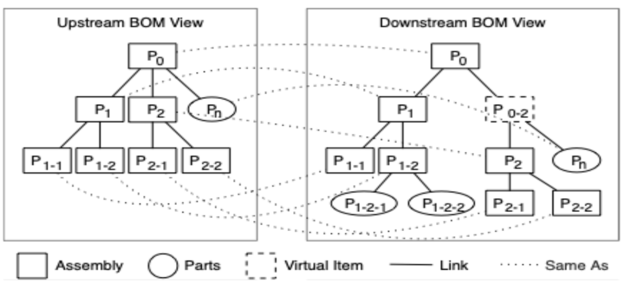

Fig. 7. Schematic diagram of the hierarchical model of traceability relationship

Fig. 7 illustrates a dual-layer traceability model for the BOM view in the ETO-oriented mode. In a highly customized production mode such as ETO, the product structure often requires engineering changes according to customer requirements, so the management of relationship traceability is particularly important. The Upstream BOM View on the left side and the Downstream BOM View on the right side of the figure represent the views of the product structure at different stages. Vertically, each of the two views takes P0 as the root node and adopts a tree hierarchical structure, which reflects the change traceability mechanism within the same view and ensures that the version update dependency relationship can be traced when engineering changes are made. Horizontally, the upstream and downstream BOM views are connected by dotted lines, demonstrating the collaborative traceability relationship among different views, which is especially important for frequent engineering changes and quick response to customer needs in ETO mode.

This dual-layer traceability model can effectively support engineering change management in ETO mode, ensuring rigor of version management through vertical change traceability, and synchronous delivery of changes between different views through horizontal collaborative traceability, thereby establishing a complete traceability system.

3.2.2 Change traceability definitions

In order to more accurately characterize the history of changes in the product structure, we define change traceability relationships in the BOM change traceability model. We use 2-tuple to represent such traceability relationships, which are both intuitive and flexible. For change traceability of material revisions, given a material m, Its set of revisions is $V(m)$, and we define the change traceability mapping function $Change_m$ as:

$$Change_m: V(m) \to V(m) \cup \{\emptyset\}$$

This indicates that the change traceability of a material revision can be null, and the function mapping of a change traceability can only be mapped to its predecessor revision, i.e.:

$$\forall m \in V(m),\ Change_m(m) \neq \emptyset \Rightarrow t\big(Change_m(m)\big) < t(m)$$

The function t is used to get the timestamp when the material revision is created. The change traceability relationship applies not only to material revisions, but also to links. In the links set R, we define a similar change traceability mapping function:

$$Change_r: R \to R \cup \{\emptyset\}$$

A link's change traceability mapping can also only be mapped to its predecessor link. A change traceability mapping function exists for a versions collection S_w which is under the same view w:

$$Change_s: S_w \to S_w \cup \{\emptyset\}$$

3.2.3 Definition of collaborative traceability

There is a horizontal collaborative traceability between different views of the product structure. For links in two different views, we define the collaborative mapping function:

$$coop_r: R \to R \cup \{\emptyset\}$$

Material revisions, as basic elements in the product structure, are consistent across all views, i.e., a material revision is the same version in any view, and this uniformity eliminates any extra need for additional co-tracing relationships for material revisions.

For commit versions in views a and b, there are synergistic traceability sets:

$$coop_s: S_a \to S_b \cup \{\emptyset\}$$

This mapping relationship indicates that view a is derived from view b, and view b exists before view a in the time series. This traceability mechanism is designed to significantly enhance the ability to manage the product structure. Within a single view, the change traceability mapping function constructs a complete change history chain, making version control more accurate and reliable. Across multiple views, the collaborative traceability mapping function builds a bridge between views. For example, changes in the engineering view can be accurately passed to the manufacturing view. With the recursive nature of the traceability function, the system is able to comprehensively assess the scope of impact of any change, thus effectively preventing potential risks. Based on the loose design of the co-evolutionary relationship function, different views can maintain relative independence and realize parallel development without losing the correlation among them. This multi-level traceability system not only ensures the consistency of product data, but also provides a flexible support framework for collaborative design.

3.3 ETO-oriented BOM change operation

In ETO mode, BOM changes are characterized by multiple sources, correlation and complexity. In order to effectively manage these changes, it is necessary to establish a systematic change operation mechanism. In this section, the ETO-oriented BOM change operation system is constructed from two change dimensions: BOM operation and view operation.

3.3.1BOM operation

BOM operations are mainly oriented towards the process of BOM changes that occur within the same view. This process occurs when the version of the material and the structure of the links change within the view, and these changes form a new commission version of the view. We establish a parent relationship from the new version to its previous version to record the change process.

In the BOM structure definition, multiple views can share material and link data, therefore the changes within a view are actually operations on the elements in the material version set and link set under the commission version of the view. In addition, since materials as an enterprise master data does not usually have different definitions at different stages, its definition only changes globally in the enterprise. Based on the above operation logic, we have defined the following BOM operation.

We define the material change operator $OP_{change\_material}$ to perform material update operations:

$$OP_{change\_material}: (m_{previous}, m_{next}) \rightarrow m_{next}$$

where $m_{previous}$ denotes the current revision of material m_i, i.e., $m_{pre} \in V_i$. m_{next} is the updated revision of material m_i. This operator adds the material revision m_{next} to the set of material revisions and establishes a parent-child relationship between the old and new versions, i.e., $m_{previous}$ is set to be the parent of m_{next}: $V_i = V_i \cup \{m_{next}\}$, $p_m(m_{next}) = m_{previous}$. In addition, we define the material add operator $OP_{add\_material}$ to represent the process of adding an existing material to the views:

$$OP_{add\_material}: (w_i, V_{new}) \rightarrow w_i'$$

where w_i denotes the current view and V_{new} denotes the set of new material revisions to be added. If the edit commission of the current view w_i is $s_{i\bullet k} = (V_{i\bullet k}, R_{i\bullet k})$, then the new material revisions to be added to the current view commission can be expressed as: $s_{i\bullet k} = (V_{i\bullet k} \cup V_{new}, R_{i\bullet k})$. During the view revision process, it is also necessary to configure a new link relationship for the material, and this operation is expressed by the link relationship addition operator $OP_{add_relation}$:

$$OP_{add\_relation}: (w_i, V_{input}, V_{output}) \rightarrow w_i'$$

where w_i is the current view, V_{input} is the set of input material revisions, and V_{output} is the set of output material revisions. Thus, in the current commission revision $s_{i\bullet j}$ of view w_i, the linking relationship between materials is established by operator $OP_{add\_relation}$, i.e., $s_{i\bullet j} = \left(V_{i\bullet j}, R_{i\bullet j} \cup \{(V_{input}, V_{output})\}\right)$. When it is necessary to configure a new input material V_{input} or output material V_{output} for an existing link r, we use operator $OP_{update\_relation}$ to represent it:

$$OP_{update\_relation}: (r, V_{input}, V_{output}) \rightarrow r'$$

When the link is updated, we get the new link relationship $r' = (V_{input}, V_{output})$. This update operation needs to satisfy the following constraints: the input and output material revisions must already exist in the material of the current version of the view, the two sets must not be null, and there must be no intersection between them. If the desired material revision does not exist in the view, then the $OPadd_material$ operation needs to be performed first to add the material. If the number of material revisions in the new link r' is less than the number in the original link r, then there is a difference set $\Delta_V = V(r) \setminus V(r') \neq \emptyset$. If in Δ_V, there is a material set $Vdis$ that is no longer referenced by the other material links, then the included materials need to be removed from the view by the operation $OP_{remove_material}$:

$$OP_{remove\_material}: (w_i, V_{dis}) \rightarrow w_i'$$

During this operation, the material is removed from the current commission version of the view to perform operation $s_{i\bullet j} = \left(V_{i\bullet j} \setminus V_{remo}, R_{i\bullet j}\right)$. In practical applications, some link relationships may become unnecessary, which is common in the purchase view—for example, the purchaser does not need to pay attention to the specific process of standard parts. For such operations that require the removal of a link relationship, we use the operator $OP_{remove\_relation}$:

$$OP_{remove\_relation}: (w_i, r_{dis}) \rightarrow w_i'$$

When a link removal operation is performed, the corresponding link is removed from the view commission, i.e., $w_{i\bullet j} = \left(V_{i\bullet j}, R_{i\bullet j} \setminus \{r_{dis}\}\right)$. After the link is removed, additional cleanup is required: the input and output materials that were originally associated with the link are examined, and if they are no longer referenced by other links in the view commission, they need to be removed from the view commission via the $OP_{remove\_material}(w_i, V_{dis})$, where V_{dis} represents the set of materials that have lost their association.

3.3.2 View Operations

There is an evolution between multiple views of the BOM, and we use the upstream relationship to record the dependency of the two views before and after the evolution, i.e., the original view is the upstream view from which the new view is evolved. Similarly, we use the upstream relationship to record the link changes that occur across views. Unlike the parent relationship, the upstream relationship is not a strict chronological update relationship, and the upstream and downstream elements are not comparable, but only have different meanings in different views.

In the ETO mode, the BOM and its views go through multiple evolution stages as the project progresses, each involving different operational requirements. These evolutions can be classified into two types of basic management operations: view construction management and version state management. Among them, view construction management is the process of building the basic framework of the BOM, including the creation of empty views, building views based on existing materials and linking relationships, and deriving new views based on existing ones. This is especially critical in ETO projects, as each customization project may require the creation of specific engineering views to meet unique customer requirements. Version state management, on the other hand, is a key node for controlling view state transitions, marking the transitions of a view from a published state to an edited state, and documenting these changes by creating new view versions. This management ensures traceability of each customized solution and is critical for change management and version tracking of ETO projects. These basic operations can be combined to form more complex operational processes to meet the various needs in an ETO project. We have standardized the definition of these operations based on the BOM version definition model, which not only ensures the consistency of the operations, but also provides a reliable database for product customization under the ETO mode.

In terms of view creation, we define three basic operators. The first is the empty view creation operator $OP_{create\_empty_view}$:

$$OP_{create\_empty_view}: k \rightarrow w_i$$

where k is the identifier of the new view, $w_i = (s_{i\bullet 1})$, $s_{i\bullet 1} = (\emptyset, \emptyset)$. This operator is used to create an empty view without any items and links. The next operator OP_{create_view} is to create a view containing the initial content:

$$OP_{create\_view}: (k, V_{init}, R_{init}) \rightarrow w_i$$

where k is the identifier of the new view, V_{init} is the set of initial revisions of material, and R_{init} is the set of initial links. The newly created view is structured as $w_i = (s_{i\bullet 1})$, where $s_{i\bullet 1} = (V_{init}, R_{init})$. The third is a replication operator OP_{copy_view} that supports view reuse:

$$OP_{copy\_view}: (k, w_i) \rightarrow w_j$$

where k is the identifier of the new view and w_i is the existing view. This operator realizes data sharing by pointing the current commit version of the new view k to the latest commit version $s_{i\bullet k} = (V_{i\bullet k}, R_{i\bullet k})$ of the existing view w_i, which effectively reduces data redundancy.

Version state management is the cornerstone of BOM view change management. It ensures traceability and consistency of product structure evolution. Versioning operations, the fundamental path to achieving this management, provide key functions such as creation, comparison, and back-rolling, enabling engineers to effectively manage and coordinate complex product structure changes. The view exists in a release state, which is a static state that allows anyone to see the same BOM structure at any time. In an ETO project, when engineers need to make changes to a published customized product view, we do not directly modify the published version of the view, but instead create a new engineering version for the changes. When modifying a commission version of the same view, the two commissions before and after creation belong to the same view, and we define the view derivation operator $OP_{derivative}$ for this operation:

$$OP_{derivative}: w_i \rightarrow w_i{}'$$

Assuming that the latest commit version of view w_i before derivation is s, this operation creates a new commit version that contains the material and linking relationship of the original commit version, $s' = (s.V, s.R)$. A vertical update dependency parent relationship $Change_p(s') = s$ is established between the old and new commit versions, indicating that s' is derived from s. When the derivation is complete, the new version is added to the view: $w_i' = w_i \cup \{s'\}$.

Fig. 8 depicts the change process of a commit version of a view. Subfigure(a) depicts the structure of a commit version with a number of materials and their links. Subfigure(b) depicts the changed state of the view, with a new commit version and a parent relationship pointing back to the original commit version and reusing the materials and links that have not changed. In the new commit version, material m_{1*2} and link r_1 are removed, material m_{2*1} is updated to version m_{2*2}, link r_2 is updated to link r_{2*}, and link r_{new} is added.

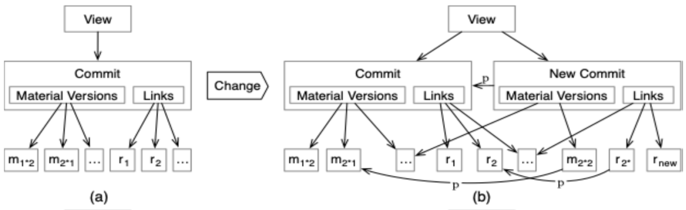

Fig. 8. Change process for the view commission version

The second scenario deals with derivation when collaborating across views. When a commit version exists in both the current and its collaborating view, i.e., $(s \in w_i) \wedge \left(s \in coop_{up}(w_i)\right)$, we use the view derivation operator OP_{branch}:

$OP_{branch}: w_i \to w_i'$

In this case, the system creates a full copy$s' = copy(s)$ of the commit version s and establishes an upstream-downstream co-evolutionary relationship between the two commits $coop_{up}(s') = s$. This design ensures that different views can evolve independently while maintaining necessary co-evolutionary relationships. When the link relationship is subsequently modified using operations such as $OP_{add_relation}$, the relationship between the link before and after the modification needs to be configured according to the relationship between the commit version to which the link belonged before modification and the version currently being edited.

Fig. 9 depicts the process of changing upstream and downstream views. In particular, subfigure(a) depicts a commit version having several materials and links; subfigure(b) depicts the structure after the change in the new view. The new view forms a parent relationship with the original view. An upstream relationship is formed between the commission versions in the new view and in the original view, and the changed links r_{2*} in the new view form the same upstream relationship with the original links r_2 across the views.

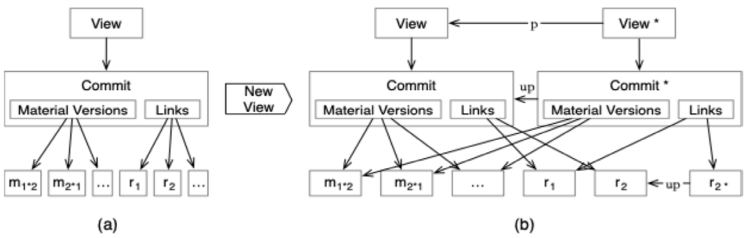

Fig. 9. Change process for a new view

4. Model application

We applied the BOM change control model to a large equipment manufacturer that utilizes a typical ETO mode. The enterprise focuses on providing total solutions for industrial factories, and the projects are highly customized, usually containing tens of thousands of material items and involving deep integration of multiple disciplines such as mechanical, electrical, and automation. However, the complexity in the ETO model increases significantly, and enterprises face problems such as frequent design changes, serious data silos, and inefficient supply chain collaboration, all of which ultimately lead to delayed delivery, seriously affecting customer satisfaction and enterprise competitiveness. In the context of digital transformation, the enterprise urgently needs to optimize BOM change management to cope with the challenges of fragmented data, lagging change response and difficult supply chain management, so as to fundamentally shorten the delivery cycle and improve overall operational efficiency and customer delivery satisfaction.

We implemented the BOM change control model proposed in this paper to develop a product change management system for this organization. Fig. 10 shows the system architecture and the BOM management and change process. The participants include BOM engineers, design engineers, suppliers, and the IT system. BOM engineers are responsible for the creation and management of the bill of materials and the synchronization among multiple views (e.g., EBOM, MBOM, PBOM). Design engineers provide design change information and verify the technical feasibility. Suppliers adjust supply chain plans and provide real-time material delivery status in response to the demand for BOM changes. The IT system supports data storage and delivery status through seamless integration with ERP/SAP.

In terms of application, several components work together. BOM Intelligence utilizes AI to analyze the impact of changes and give suggestions for plan adjustments. Change Portal is used for submitting and tracking change requests. The Data Inspector is responsible for data verification. Message Robot automatically notifies relevant parties and triggers collaborative processes. The core process includes change identification, impact analysis and recommendation, implementation and validation, and notification and collaboration to ensure the accuracy and consistency of changes. The knowledge base covers BOM data domain, change domain, supplier domain, and project domain, supporting multi-view data storage, change lifecycle management, supplier performance evaluation and cross-project data sharing, which comprehensively improves the enterprise's digital capability and supply chain collaboration efficiency.

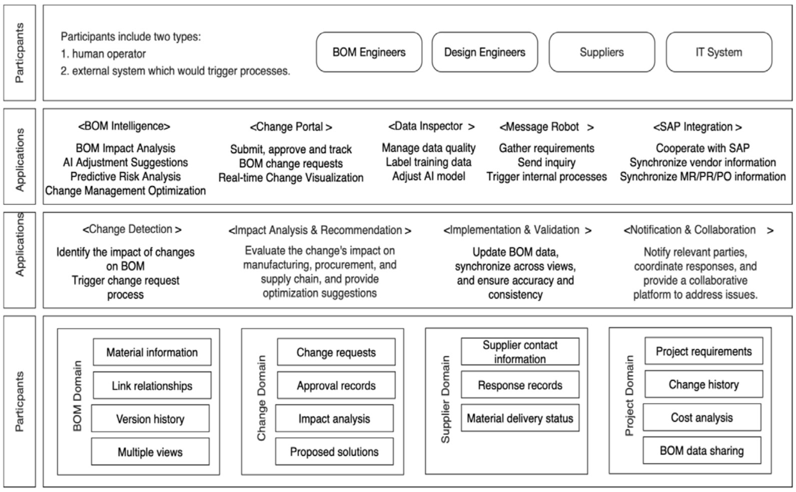

Fig. 10. Application architecture

In the BOM view change traceability process, the core steps can be divided into the following parts. The first step is to change identification and recording. When there is a change in design or requirements, the system automatically generates a change request and associates it with related views (e.g., EBOM, MBOM, PBOM). The change identification module clarifies the source of the change (e.g., design department or supply chain feedback) as well as the specific layers and material items affected. This step ensures the completeness and traceability of the change information. The second step is multi-view synchronization and data validation. Before the change is implemented, BOM engineers utilize a system tool to verify the consistency of data among EBOM, MBOM, and PBOM. The system automatically detects data conflicts or omissions and ensures information synchronization across different views. This not only improves data accuracy, but also reduces subsequent problems caused by inconsistent information. The third step is change impact analysis. The system analyzes the scope of the change through the AI module, including the impact of assembly relationships, adjustments to material requirements, and updates to supply chain delivery plans. The fourth step is implementation and validation, where the change is formally applied to specific design and production processes. Design engineers verify the technical feasibility of the changes, supply chain departments adjust material delivery plans, and production departments adjust processes according to the new BOM. The system tracks the progress of the changes in real-time to ensure that each step of the process is executed according to the plan. Throughout the process, the system notifies relevant parties (e.g., suppliers, production line managers) and triggers a collaborative process to ensure that all participants have a clear understanding of the change content and timeline. After the change is completed, the system generates a detailed change traceability report, including the time, participants, impact area, and final result of the change.

Table 1 shows the sample BOM data of an enterprise project, which is the basic data for change management. This system unifies the control of BOM data at each stage in the product life cycle. Fig. 11 gives the connection relationship of the BOM structure in the transition process of multiple views. Most of the connections in the figure are reused by EBOM, PBOM, and MBOM, and when PBOM and MBOM make additions to the design, they only need to create the connections that they have changed.

When the quantity of the material "Squeeze roll unit" (material number 0304010100) is adjusted from 1 to 2 in the mounting position 0010.0001.0002.0001, the system automatically recognizes the change and analyzes its impact on the upper assembly and the lower component. It then collaborates on the relevant views through the process described above. The system is responsible for the consistency of data across views and ensures that the change is implemented and verified without error, while adjusting the plans for each stage of the production and supply chain.

Through this series of processes, enterprises can not only realize the full lifecycle management of BOM changes but also improve the response speed and collaborative efficiency of complex projects, so as to better cope with the challenges under the ETO model.

Table 1
BOM basic data

NO.	Level	Position	Parent Material	Material No.	Description	Quantity
1	0	10		0100000000	Coil deposit	1
2	0	10		0200000000	Strip centering control system	1
3	0	10		0300000000	Process tank assembly	1
4	1	10.04	0300000000	0301000000	Hexagon head screw	12
5	1	10.04.01	0300000000	0302000000	Washer	24
6	1	10.04.02	0300000000	0303000000	Hexagon regular nut	12
7	1	10.01	0300000000	0304000000	Process tank	1
8	2	10.01.02	0304000000	0304010000	Squeeze roll	1
9	2	10.01.03	0304000000	0304020000	Brushing machine	2
10	2	10.01.04	0304000000	0304030000	Locknut	4
11	2	10.01.04.01	0304000000	0301000000	Hexagon head screw	8
12	3	10.01.02.01	0304010000	0304010100	Squeeze roll unit	1
13	3	10.01.02.02	0304010000	0304010100	Squeeze roll unit	1
14	4	10.01.02.01.06	0304010100	0304010101	Flange	1
15	4	10.01.02.01.09	0304010100	0304010102	Guide	4
16	4	10.01.02.01.10	0304010100	0304010103	Plate	4
17	4	10.01.02.01.11	0304010100	0304010104	Scale	1

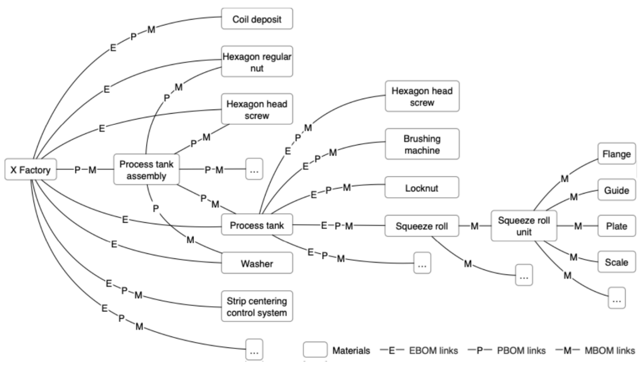

Fig. 11. Passing of BOM structure in multiple views

Through the application of the model, the enterprise has achieved significant results in project execution and delivery efficiency. Before the implementation of the system, the enterprise faced serious delivery delays. As shown in Table 2, 80% of the parts were delayed by 60 days. The delay not only affected customer satisfaction, but also put significant pressure on the enterprise’s internal production planning and resource allocation.

Table 2
Before system implementation

PO No.	Item Name	Delay (Days)	Qty	Contractual Delivery Time
3200452379	Pendulum shear	60	1	2023/10/27
3200452379	Downcoiler housing	37	1	2023/9/18
3200452379	Machine piping	89	1	2023/11/3
3200452379	Mill housing A1	76	1	2023/10/17
3200452379	Mill housing A2	63	1	2023/9/29
3200461465	Coiler drive base	73	1	2023/3/18
3200461465	Coiler track way	78	1	2023/3/5
3200461465	Edger housing	62	1	2023/5/19
3200461465	Downcoiler housing B1	59	1	2023/9/5
3200461465	Downcoiler housing B2	62	1	2023/6/7

After the introduction of the product change management system, the enterprise’s project management capability has been comprehensively improved. According to the data in Table 3, after the implementation of the system, most of the components of the procurement contracts were delivered on time. Even if there were slight delays in individual parts, the overall delivery progress is still above 90%, which is significantly better than the status before the system was deployed.

This improvement demonstrates that the BOM change management model effectively enhances the enterprise's ability to respond to changes, optimizes supply chain collaboration efficiency, and significantly shortens the project delivery cycle through real-time monitoring and dynamic adjustments. These benefits ultimately achieve the goal of improving customer satisfaction and enhancing market competitiveness.

Table 3
After system implementation

PO No.	Item Name	Delay (Days)	Qty	Contractual Delivery Time
3200500934	Spindle head support F1	0	1	2024/11/7
3200500934	Spindle head support F2	5	1	2024/11/7
3200500934	Spindle head support F3	0	1	2024/11/7
3200500934	Base plate G1	6	1	2024/10/13
3200500934	Base plate G2	0	1	2024/10/13
3200500934	Base plate coiler	0	1	2024/10/13
3200525618	Roller table bridge with side guides	0	1	2024/9/24
3200525618	Roller table with funnel guide	7	1	2024/9/24
3200525618	Roller table with side guide	0	1	2024/9/24
3200525618	Coil Stripper Car H1	3	1	2024/9/5
3200525618	Coil Stripper Car H2	0	1	2024/9/5
3200525618	Coil Stripper Car H3	0	1	2024/9/5

5. Conclusion

The ETO mode is of great significance in the modern manufacturing industry. Its highly customized features satisfy customers' personalized needs, but it also brings management challenges such as frequent changes, complex collaboration, and high risks. To address these issues, we propose a dual-layer BOM-based change control model in this paper.

Firstly, we develop a BOM model definition based on version control. This approach constructs a generalized BOM structure system by introducing material revisions, material relationship links, and a multi-view mechanism. These elements enable model definition and change representation throughout the full lifecycle of the product.

Secondly, we construct a dual-layer change traceability model using vertical version chains and horizontal view collaborative traceability. This model ensures change traceability and cross-view consistency of product structure in ETO mode.

Subsequently, we established an ETO-oriented BOM change operation model. This model achieves efficient change management and full lifecycle traceability of product structure through standardized intra-view BOM change operations and cross-view cooperative operations.

Finally, practical application in a large equipment manufacturing enterprise verifies the model's significant effects. These include improved change response efficiency, optimized management processes, and enhanced digital transformation support. These benefits provide reliable support for supply chain collaboration and operational efficiency improvement in ETO enterprises.

Based on the results of this research, future work will expand the potential of the BOM change model in the ETO mode. Key research directions include:

(1) Developing a deep integration model of BOM and WBS (Work Breakdown Structure). This will realize real-time linkage between product structure changes and project task nodes by constructing a change-driven dynamic mapping mechanism, ensuring accurate transmission of engineering changes to resource scheduling and progress planning.

(2) Developing knowledge mapping technology for change traceability. This approach will model historical change records, process parameters, quality data, and other multivariate information, providing intelligent early warning support for change decision-making.

Declarations

Funding

This work was supported by the Natural Science Foundation of Sichuan Province, China (No. 2023NSFSC1010); and the Humanities and Social Sciences Research Project of the Ministry of Education, China (No. 20YJC630146).

Competing Interests

The authors have no relevant financial or non-financial interests to disclose.

Ethics approval and consent to participate

All authors declare that this article does not have any academic ethics issues and strictly follows the journal submission rules and agree to participate in the research work of this paper and publish it in the International Journal of Advanced Manufacturing Technology.

Consent for publication

All authors agree to publish this article in the International Journal of Advanced Manufacturing Technology.

Author contribution

All authors contributed to the study conception and design. Material preparation, data collection and analysis were performed by [Chunhua Wan] and [Ji Ma]. The first draft of the manuscript was written by [Chunhua Wan] and all authors commented on previous versions of the manuscript. All authors read and approved the final manuscript.

References

Bhalla, S., Alfnes, E., & Hvolby, H. H. (2023). Tools and practices for tactical delivery date setting in engineer-to-order environments: a systematic literature review. *International Journal of Production Research, 61*(7), 2339-2371.

Chowdhury, A. M., & Moon, S. (2023). Generating integrated bill of materials using mask R-CNN artificial intelligence model. *Automation in Construction, 145*, 104644.

Fa, J., Wen, Y., Yongan, Z., Rong, Z., & **uhui, G. (2020, April). Dynamic BOM construction technology and application based on MBD. *In Journal of Physics: Conference Series* (Vol. 1486, No. 7, p. 072050). IOP Publishing.

Gosling, J., & Naim, M. M. (2009). Engineer-to-order supply chain management: A literature review and research agenda. *International journal of production economics, 122*(2), 741-754.

Hicks, C., McGovern, T., & Earl, C. F. (2000). Supply chain management: A strategic issue in engineer to order manufacturing. *International journal of production economics, 65*(2), 179-190.

Jing, Z., Hu, N., Song, Y., Song, B., Gu, C., & Pan, L. (2022). On the design and implementation of a blockchain-based data management system for ETO manufacturing. *Applied Sciences, 12*(18), 9184.

Kalantari, M., Rabbani, M., & Ebadian, M. (2011). A decision support system for order acceptance/rejection in hybrid MTS/MTO production systems. *Applied Mathematical Modelling, 35*(3), 1363-1377.

Lee, J. H., Kim, S. H., & Lee, K. (2012). Integration of evolutional BOMs for design of ship outfitting equipment. *Computer-Aided Design, 44*(3), 253-273.

Love, P. E., Matthews, J., & Ika, L. A. (2024). Rework in relational engineer-to-order production systems: An 'error-as-process' archetype. *International Journal of Production Economics, 274*, 109310.

Mello, M. H., Strandhagen, J. O., & Alfnes, E. (2015). Analyzing the factors affecting coordination in engineer-to-order supply chain. *International Journal of Operations & Production Management, 35*(7), 1005-1031.

Pracucci, A. (2024). Designing Digital Twin with IoT and AI in warehouse to support optimization and safety in Engineer-to-Order manufacturing process for prefabricated building products. *Applied Sciences, 14*(15), 6835.

Qiao, L., Zhang, Z., & Huang, Z. (2021). Construction of multi-project network planning based on bom and its resource leveling. *IEEE Access, 9,* 71887-71899.

Schramm, S., Pieper, M., & Vogl, S. (2023). Orthogonal Procrustes and Machine Learning: Predicting Bill of Materials errors on time. *Computers & Industrial Engineering, 185*, 109606.

Shiau, J. Y. (2020). A semi-automatic planning BOM generator based on VMI forecasting. *International Journal of Systems Science: Operations & Logistics, 7*(2), 121-132.

Thajudeen, S., Elgh, F., & Lennartsson, M. (2022). Supporting the reuse of design assets in ETO-based components—A case study from an industrialised post and beam building system. *Buildings, 12*(1), 70.

Vaagen, H., Kaut, M., & Wallace, S. W. (2017). The impact of design uncertainty in engineer-to-order project planning. *European Journal of Operational Research, 261*(3), 1098-1109.

Wang, S., Liu, X., Bai, Z., & Xiao, J. (2023). A BOM model transformation method for hierarchical production planning management process of complex products. *Advanced Engineering Informatics, 58*, 102138.

Wang, Y., Ren, W., Zhang, C., & Zhao, X. (2022). Bill of material consistency reconstruction method for complex products driven by digital twin. *The International Journal of Advanced Manufacturing Technology, 120*(1), 185-202.

Wang, Y., Wang, Y., Ren, W., & Jiang, Z. (2024). Knowledge driven multiview bill of material reconfiguration for complex products in the digital twin workshop. *The International Journal of Advanced Manufacturing Technology, 130*(7), 3469-3480.

Willner, O., Gosling, J., & Schönsleben, P. (2016). Establishing a maturity model for design automation in sales-delivery processes of ETO products. *Computers in Industry, 82*, 57-68.

Zhao, X., Wei, S., Ren, S., Cai, W., & Zhang, Y. (2024). Integrating MBD with BOM for consistent data transformation during lifecycle synergetic decision-making of complex products. *Advanced Engineering Informatics, 61*, 102491.

Zhou, C., Liu, X., Xue, F., Bo, H., & Li, K. (2018). Research on static service BOM transformation for complex products. *Advanced engineering informatics, 36*, 146-162.