Academic Journal of Science and Technology
ISSN: 2771-3032 | Vol. 19, No. 1, 2026

Research on a MES-Based Unified Code Generation, Management, and Traceability System for Discrete Manufacturing

Yongchuan Ren [2], Kaihuai Luo [1], Feng Liu [2], Guofeng Xie [2], Tianxing Chen [1, *]

[1] College of Mechanical Engineering, Southwest Jiaotong University, Chengdu, Sichuan, China
[2] Civil Aviation Logistics Technology Co., Ltd., Chengdu, Sichuan, China
[*] **Corresponding author**: Tianxing Chen (Email: txchen@swjtu.cn)

Abstract: To address common challenges faced by discrete manufacturing enterprises during digital and intelligent transformation—such as inconsistent material coding conventions, difficulties in coordinating multi-source BOM data, and disruptions in end-to-end lifecycle traceability—this paper proposes an MES-based framework for unified code generation, management, and traceability. The proposed framework adopts a bidirectionally linked “one item, two codes” strategy and establishes a structured model consisting of a static material code and a dynamic batch code. Through configurable coding rules, the framework enables unique identification of object attributes and dynamic mapping to production batches. By integrating a rule-parsing engine with a full-chain indexing mechanism, the system achieves closed-loop process management spanning BOM import, automatic code assignment, and shop-floor execution. Engineering practice demonstrates that the proposed system effectively mitigates conflicts caused by multiple codes for the same item and significantly improves material identification efficiency and quality traceability accuracy in complex manufacturing scenarios.

Keywords: Coding Rules; Unified Coding; Digital and Intelligent Transformation; Manufacturing Execution System (MES); Code-based Traceability.

1. Introduction

With the continued advancement of information technologies, digital and intelligent manufacturing has become a key driving force for the transformation, upgrading, and high-quality development of the manufacturing industry. Future manufacturing paradigms are expected to evolve toward greater intelligence, connectivity, and sustainability. Emerging technologies such as artificial intelligence and the Internet of Things are being deeply integrated into manufacturing systems to enable efficient, intelligent, and sustainable manufacturing ecosystems [1].

In the context of smart manufacturing, structured, modular, and parametric modeling has increasingly been recognized as a set of core principles. “Structured” modeling emphasizes standardized decomposition of objects, processes, and data; “modular” modeling focuses on capability reuse and composability; and “parametric” modeling expresses business constraints through computable rules [2]. Within manufacturing execution systems (MES), these principles are particularly critical for coding management.

In discrete manufacturing, especially for large and complex electromechanical equipment, products are typically composed of thousands of parts and components assembled through sophisticated processes [3]. Compared with process manufacturing, discrete manufacturing is characterized by deep hierarchical bill of materials (BOM), non-continuous process routes, frequent engineering changes, and high-mix low-volume production. These characteristics inherently create a data gap between the design domain and the manufacturing domain. In practice, the absence of unified coding standards often leads to issues such as “multiple codes for the same item” or “items without codes” when transforming a design BOM into a manufacturing BOM, which severely constrains cross-department collaboration and end-to-end quality traceability.

To address these challenges, this paper proposes an MES-oriented solution for unified code generation, management, and traceability. First, based on a structured modeling approach, a bidirectionally linked “one item, two codes” strategy is designed to abstract coding objects, business processes, and data in a unified manner, thereby forming a closed-loop business workflow covering planning, execution, quality inspection, and traceability. Second, at the system architecture level, a set of modular service components—including rule parsing, sequence control, and traceability indexing—is developed to support flexible configuration and rapid system evolution. Finally, engineering practice is conducted to validate the effectiveness of the proposed system in improving material identification efficiency and enhancing the accuracy of quality traceability, aiming to help discrete manufacturing enterprises build an efficient and scalable digital-intelligent coding system.

2. Design Principles for Unified Coding

In discrete manufacturing enterprises, cross-department and cross-plant collaboration often suffers from inconsistent coding conventions, fragmented rule sets, frequent identifier conflicts, and non-traceable version iterations [4]. Meanwhile, the diversity of BOM structures and data sources can lead to “multiple codes for the same item” or “items without codes,” which breaks the data chain among planning, execution, quality inspection, and after-sales service. Therefore, it is necessary to establish a unified coding scheme for materials and related objects across the enterprise production domain [5]. The design of coding rules should follow both fundamental principles and evolutionary principles.

2.1. Fundamental Principles

(1) Uniqueness: Within a predefined scope, at any time, a given code value shall correspond to one and only one entity instance. Moreover, the code must not be reused or become ambiguous throughout its lifecycle.

(2) Stability: Coding rules should not be arbitrarily modified due to changes such as the addition or removal of materials. Once a coding rule is altered, it may introduce incompatibilities across the system and lead to cascading management issues; therefore, rule changes should be strictly controlled.

2.2. Evolutionary Principles

(1) Extensibility: As enterprises evolve, the material set and its structure will inevitably change. Material coding should not be designed solely for the current material catalog but must also accommodate future expansion needs [6]. The coding scheme should reserve sufficient code space and support scalable allocation to meet continuously growing material demands.

(2) Enterprise-wide uniformity: A single unified material coding scheme should be adopted across the entire enterprise, including subsidiaries and functional departments, to enable consistent information sharing and efficient material dispatching and coordination [7].

(3) Simplicity: Codes should be as concise as possible, with an appropriate length, to reduce storage overhead and minimize the probability of coding errors during manual handling and system integration.

3. Coding Rule Engine and Coding System Architecture

To gain an in-depth understanding of the practical challenges associated with coding management on the shop floor, this study conducted an on-site investigation at Civil Aviation Chengdu Logistics Technology Co., Ltd. Although the company has deployed an MES for production management, inconsistent coding conventions and fragmented rule definitions have resulted in low inter-system collaboration efficiency and poor data consistency. Consequently, the existing system landscape is unable to effectively support comprehensive quality traceability and enterprise-wide data sharing.

This work first examined the company's material taxonomy. In practice, the company classifies materials into products, purchased products, semi-finished products, raw materials, auxiliary materials, standard parts, general-purpose parts, fixed assets, and office supplies. Prior to defining a unified coding scheme, the material taxonomy was refined and delineated in detail to facilitate the subsequent establishment of enterprise-wide coding rules. The company's product and material coding taxonomy are illustrated in Figure 1.

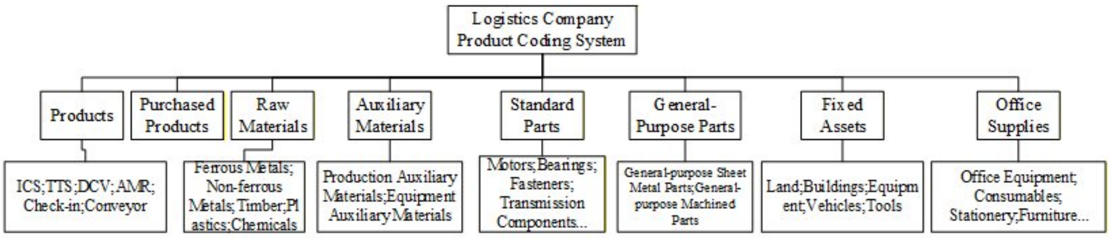

Figure 1. Material coding taxonomy

3.1. Establishment of the Unified Coding Scheme

Conventional coding schemes often attempt to embed all material information into a single long identifier. Such an approach violates the orthogonality principle in data governance, as it couples heterogeneous attributes into one representation and increases the cost of evolution and maintenance. To resolve the inherent coupling between static attributes and dynamic attributes in discrete manufacturing, this paper proposes a "one item, two codes" strategy. In essence, it is a decoupled model based on object-oriented thinking, where a static class definition is separated from a dynamic instance mapping [8]. Specifically, the material code represents the static characteristics of an item—such as category, attributes, structure, and process—at the enterprise level and serves as a cross-system, cross-lifecycle primary key. The batch code carries dynamic production information and functions as a key field for batch traceability and quality management.

3.1.1. Material Code

The material code is used to establish material master data records and to distinguish different material types. The design rules are as follows:

(1) Code structure: The material code adopts a five-level structure, organized into two segments: a classification segment and a serial-number segment, with a total length of 12 digits.

(2) Classification segment: The material classification consists of four hierarchical levels, corresponding to:

Level 1: material category, Level 2: attribute, Level 3: structure, Level 4: process. Each level uses two digits. The value range for Levels 1–3 is "01–99", while Level 4 ranges from "00–99."

(3) Serial-number segment: The serial number uses four digits to uniquely identify different instances under the same classification. It is assigned in production sequence order, ranging from "0001" to "9999."

This scheme ensures uniqueness, extensibility, and recognizability, while providing clear hierarchical semantics. For example, the material code for a non-standard straight conveyor equipment is 10-32-31-01-0001. This indicates that the item is a product (Level 1), its attribute is "conveyor" (Level 2), its structure is "straight conveyor" (Level 3), and its process/part type is "assembly" (Level 4), thereby capturing complete semantics from category to process.

Table 1. Unified material coding scheme

Level 1 code (01-99)	Level 2 code (01-99)	Level 3 code (01-99)	Level 4 code (00-99)	Serial No. (0001-9999)
xx	xx	xx	xx	xxxx
10- Product; 11- Purchased product; 30- Raw material; 40- Auxiliary material; 50- Standard part; 60- General-purpose part; 80- Fixed asset; 90- Office item	11-ICS,12-TTS,13-DCV,14-AMR,31-Check-in; 32- Conveyor; 33- Turntable; 70- Electrical control, …	01- Straight conveying equipment;21- Curved conveying equipment;11- Curved conveyor,…	01- Assembly;02- Sheet meta;03- Machining;04- Weldment;05- Molding;06- Other, …	Decimal numbering, assigned in production sequence

3.1.2. Batch Code

The batch code distinguishes different production batches of the same material arising from differences in time, source, or process routing. It is the core identifier for end-to-end tracking and quality traceability. In the proposed system, the enterprise imports a BOM list containing existing material codes into the MES. The system then automatically generates batch codes and establishes a one-to-many binding relationship between a material code and its associated batch codes.

The batch code adopts a three-part structure with a total length of 9 characters, as illustrated below:

Level 1: production batch abbreviation, Level 2: reserved field for future extension of material-related batch attributes, Serial number: incremental sequence in decimal form. For example, the batch code for a non-standard straight conveyor equipment is PH-000-0001.

Table 2. Batch coding scheme

Level 1	Level 2	Serial No. (0001–9999)
PH	000	xxxx
Production batch abbreviation	Reserved for future batch-related attributes	Decimal numbering, assigned in production sequence

3.2. MES-Based Coding System Architecture

To ensure the effective execution of coding rules and to enable end-to-end traceability, this paper develops an MES-based architecture for unified code generation and traceability, as illustrated in Figure 2. The overall architecture comprises three layers:

(1) Front-end layer. Mobile terminals are used to submit business requests and collect shop-floor data. The front-end supports operations such as BOM import, batch list recognition, and material information entry.

(2) Business service layer. Serving as the core logic layer, this layer is responsible for cleansing and validating multi-source BOM data, generating codes according to the unified rules, performing deduplication, and constructing full-chain indexes linking codes with process operations, inbound warehousing, and traceability records.

(3) Data storage layer. This layer stores master-data dictionaries, business transaction details, and operation logs. It supports query and retrieval, as well as statistical reporting, thereby ensuring data persistence and traceability

The three layers collaborate through service invocations and event-notification mechanisms in a decoupled manner, enabling process continuity, coding consistency, and system extensibility. With this architecture, enterprises can manage material codes under a unified convention and support full lifecycle traceability across systems and process stages. When quality issues occur, the system can rapidly locate abnormal batches and trace them back to responsible links, facilitating efficient recall and accountability.

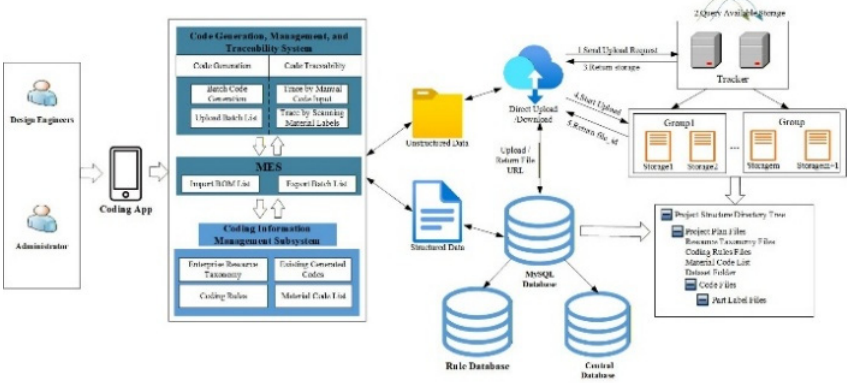

Figure 2. MES-based coding system architecture

To achieve efficient coordination among layers, the system adopts a lightweight data exchange mechanism. Between the presentation layer and the business service layer, the mobile terminal serializes the collected raw BOM data and shop-floor operation commands into JSON-formatted payloads and initiates service requests. Between the business service layer and the data storage layer, the parsing engine writes processed and structured coding data into a MySQL database, while unstructured data are uploaded to a distributed file storage system; only file index paths are retained in the database.

This bidirectional mechanism—where commands are sent upstream and data are returned downstream—combined with event-driven notifications, ensures real-time responsiveness and data consistency across the entire workflow, from code generation to traceability queries.

4. Design and Implementation

4.1. Overall System Design and Implementation

The proposed system is composed of five functional modules: system management, automatic code generation, code-based traceability, data center, and statistical analysis.

The system management module includes user management and server management. User management covers account administration and access control. Users must authenticate with valid credentials before entering the system and performing operations, and all actions are constrained by authorization policies to prevent privilege escalation [9]. Server management includes configuration and maintenance of both the database server and the file server, such as service endpoints and related connection parameters.

The automatic code generation module is the core component of the system. It mainly consists of BOM import and code generation, and batch list import. The BOM import and code generation function enables batch generation of material codes. The system parses part/material names and automatically assigns codes to each record according to predefined coding rules.

Although MES has been deployed within the enterprise, the upstream supply chain in discrete manufacturing still includes many suppliers and subcontractors with limited digitalization. As a result, source data are frequently exchanged in the form of paper-based transfer sheets, scanned PDFs, or non-standard images, creating data discontinuities during BOM import. To address these issues, the proposed module applies a multi-level data processing pipeline and key-field mapping techniques:

(1) Heterogeneous data ingestion via OCR and semantic parsing. In addition to direct import and format validation for standard BOM spreadsheets, the system introduces OCR-based processing for heterogeneous inputs. This is not a simple image-to-text conversion; instead, OCR is combined with semantic analysis to identify key feature fields—such as material name and specification—from non-standard images or scanned documents, converting them into structured text that can be processed by the system, thereby eliminating data breakpoints.

(2) Unified intermediate representation and backend mapping. After structured data are obtained, the system converts them into a unified intermediate format using JSON serialization and transmits the payload to the backend. The rule-parsing engine then applies the unified coding scheme described in Section 3, and automatically maps the material's Level 1–Level 4 classification attributes via a keyword-matching algorithm.

(3) Serial-number allocation and BOM transformation. After classification attributes are determined, the system queries the database to obtain the maximum serial number under the corresponding category and generates a unique material code according to the auto-increment logic. The generated coding information is then written back to the MySQL database, completing an intelligent transformation from the design BOM to the manufacturing BOM.

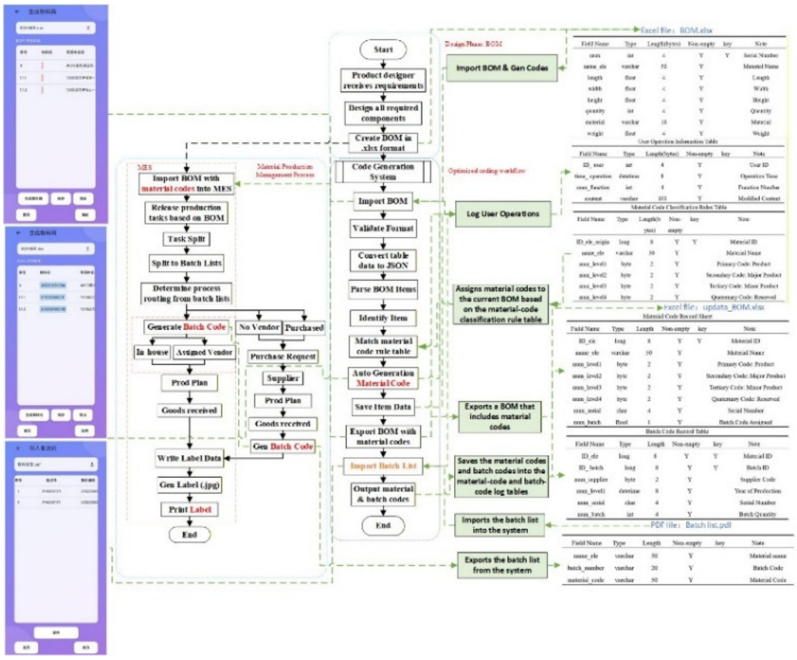

Figure 3. Workflow of automatic code generation

The code-based traceability module is a key component for enabling backward tracing and full lifecycle management. It supports fast query and localization of detailed material information, provenance records, and historical operations based on coding identifiers [10]. Users can initiate traceability requests through multiple entry points, such as manual code input or photo-based label recognition.

Upon receiving a query request, the system retrieves the corresponding data entries associated with the given code from the database and returns information such as material name, product model, production batch identifier, operator, and production time, enabling a visualized presentation of the material's end-to-end lifecycle. Furthermore, by integrating the photo recognition function, the system can automatically extract codes from on-site images and navigate users to the relevant traceability detail page, thereby providing a linked experience across "code–image–material". In practical applications, the material code traceability function can be widely used in warehousing management, quality root-cause tracing, after-sales service, and project acceptance processes, serving as an important foundation for digital-intelligent governance and visualization of coding assets.

4.2. System Testing

To evaluate the system's performance and efficiency in practical use, a comparative experiment was designed to examine the advantages of the automatic batch coding module over the conventional manual coding approach in terms of processing speed and operational efficiency.

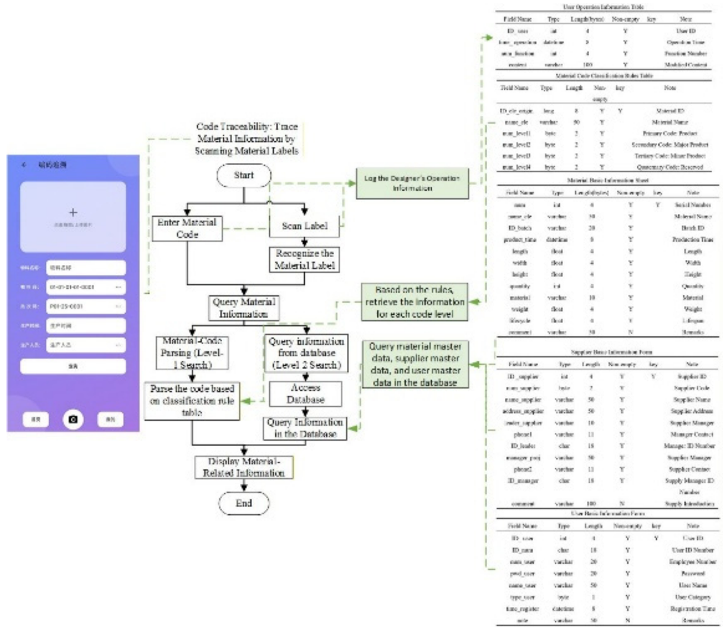

Figure 4. Workflow of code-based traceability

The test scenario used a representative material list: the BOM structure of a 1200-type straight conveyor, which contains multiple subcomponents. In the traditional approach, codes are typically created manually based on experience or reference tables. On average, it takes approximately 10 minutes to complete the coding task for a BOM list of this type. In contrast, in the proposed system, users simply import the BOM list, after which the system automatically parses the component structure according to predefined coding rules and generates the complete codes in batch. The same workload can be completed in approximately 5 seconds, substantially improving coding efficiency.

The comparison results are shown in Figure 5, demonstrating that the proposed system provides significant time savings and strong scalability for high-frequency and complex coding scenarios.

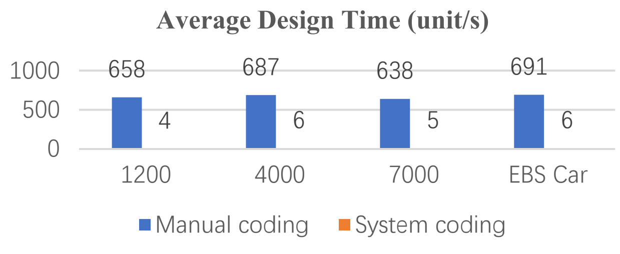

Figure 5. Comparison of average material coding time

5. Conclusion

This paper addresses common challenges in coding management and traceability for discrete manufacturing enterprises, including inconsistent coding conventions, fragmented rule definitions, and difficulties in cross-system traceability. To tackle these issues, an MES-integrated solution for code generation and traceability is proposed, combining a unified coding rule engine, an extensible data model, and systematic MES integration. Based on field investigation and a case study at Civil Aviation Chengdu Logistics Technology Co., Ltd., an MES-oriented unified coding architecture was established. A standardized 12-digit, five-level coding scheme was designed to systematically integrate product, batch, and process information.

From an architectural perspective, the system adopts a modular design and integrates mobile terminals with a lightweight MySQL database, forming a complete functional suite that covers user management, code generation, code-based traceability, a data center, and statistical analysis. Across the two core functions—automatic coding and traceability query—the system realizes closed-loop management from data ingestion and rule-driven mapping to code generation and full lifecycle backtracking. This approach improves material identification efficiency, reduces coding conflicts, and significantly shortens manual operation time.

Experimental results indicate that, compared with conventional manual coding methods, the proposed system exhibits clear advantages in processing speed, accuracy, and scalability, and is capable of supporting large-scale coding and quality traceability requirements in complex manufacturing environments. Future work will extend the solution to upstream and downstream supply chain partners to achieve end-to-end interoperability of the coding scheme and to provide foundational support for higher-level digital and intelligent transformation.

References

[1] Hu, Yasen. *Research on an Urban Rail Transit BIM Family Library Management System Based on a Unified Coding Standard* [D]. Southwest Jiaotong University, 2021. DOI:10.27414/d.cnki.gxnju.2021.003474.

[2] Wu, Simon,Lin, Sin-Jie,Jiang, Yi-Cang , et al.Bottlenecks and improvement strategies for digital transformation in manufacturing[C].//12th International Conference on Orange Technology, ICOT 2024.2024:1-4.

[3] Jewapatarakul, Darun,Ueasangkomsate, Pittawat.Digital Transformation: The Challenges for Manufacturing and Service Sectors[C].//7th International Conference on Digital Arts, Media and Technology, DAMT 2022 and 5th ECTI Northern Section Conference on Electrical, Electronics, Computer and Telecommunications Engineering, NCON 2022.2022:19-23.

[4] Jia, Tiemin. The important role of standardized material coding in enterprise electronic information systems [J]. *Electronic World*, 2017(8):51–54.

[5] Wu, Dong,Xiao, Yang.Digital Driven Global Manufacturing Industry Transition and Innovative Development[C].//2021 International Conference on Digital Society and Intelligent Systems, DSInS 2021.2021:138-142.

[6] Cai, Hong Xia,Dai, Ming Yu.Material Coding for Aircraft Manufacturing Industry[J].JOURNAL OF AEROSPACE TECHNOLOGY AND MANAGEMENT,2014,6,(2):183-191. DOI:10.5028/jatm.v6i2.315.

[7] Zhang, Wenhao, Lai, Jinrong, et al. Design and implementation of a rail transit asset management system based on unified coding [J]. *Computer Programming Skills & Maintenance*, 2022(11):107–108, 142. DOI:10. 16184/ j .cnki. comprg. 2022. 11. 020.

[8] Zhu, Li. Application practice of a material coding system in the full procurement process of traditional petrochemical warehousing enterprises [J]. *Petroleum & Petrochemical Materials Procurement*, 2023(23):4–6. DOI:10.12316/j. issn. 1674-0831.2023.23.002.

[9] Xie, Hui, Zhang, Shaodong, Meng, Shuang, Zhang, Cong, Su, Junjun, Zheng, Bin. Enterprise basic material management and information platform construction oriented to digitalization [J]. Intelligent Manufacturing, 2023(4):106–109.

[10] Ma, Xiaofei,Yue, Guanghui,Wu, Yuanchen.Application of Computer Information Technology in the Digital Transformation of Power Enterprise Management[C].//2nd IEEE International Conference on Distributed Computing and Electrical Circuits and Electronics, ICDCECE 2023.2023:1-5.