



Analyzing Multi-level BOM-Structured Event Data

Tobias Brockhoff¹ , Merih Seran Uysal¹ , Isabelle Terrier²,
Heiko Göhner², and Wil M. P. van der Aalst¹ 

¹ Process and Data Science Chair, RWTH Aachen University, Aachen, Germany

{brockhoff,uysal,wvdaalst}@pads.rwth-aachen.de

² Heidelberger Druckmaschinen AG, Heidelberg, Germany

Abstract. With the advent of Industry 4.0, increasing amounts of data on operational processes (e.g., manufacturing processes) become available. These processes can involve hundreds of different materials for a relatively small number of manufactured special-purpose machines rendering classical process discovery and analysis techniques infeasible. However, in contrast to most standard business processes, additional structural information is often available—for example, Bills of Materials (BOMs), listing the required materials, or Multi-level Manufacturing Bills of Materials (M²BOMs), which additionally show the material composition. This work investigates how structural information given by Multi-level Bills of Materials (M²BOMs) can be integrated into a top-down operational process analysis framework to improve special-purpose machine manufacturing processes. The approach is evaluated on industrial-scale printer assembly data provided by *Heidelberger Druckmaschinen AG*.

Keywords: Process Mining · Bill of Materials · Operational Processes · Industry 4.0 · Offset Printing

1 Introduction

With the advent of digitalization, data on an increasing number of processes are recorded. *Process mining* is the emerging key discipline concerned with the analysis of such data to provide insights into processes and, eventually, to improve them. Traditionally, *event data*, i.e., a set of discrete events that are linked by a certain case notion, have been recorded in business management systems, which, for example, handle *order-to-cash* or *purchase-to-pay* processes. However, with the rise of *Industry 4.0*, more and more event data from manufacturing and assembly processes become available. The analysis of these so-called *operational processes* [1] using process mining is therefore key to not only remove friction from companies' administrative workflows but also to optimize and steer their manufacturing processes.

In contrast to standard business processes, *operational* processes frequently provide additional structural information. A common approach to structure

the production, particularly for complex products, is by means of a **M**ulti-level **M**anufacturing **B**ill of **M**aterials (M²BOM) that shows the hierarchical composition of required materials. These models are for example supported by manufacturing ERP systems such as *SAP* [13]. Moreover, *operational processes* often involve a large number of materials and assembly tasks that render fully-automatic model discovery infeasible. However, model discovery plays a central role in classical process mining-based process analysis frameworks such as PM² [6], where the mining & analysis stage implementation comprises automatic model discovery, conformance checking, and model enhancement. Thus, the adaptability of standard analysis approaches to *operational processes* is limited. Thus, given *operational* event data where each assembly case is endowed with its corresponding M²BOM, we propose a two-stage refinement of PM²'s mining & analysis step. The first substage targets a general and comprehensive performance overview exploiting additional structural information; the second substage concerns the analysis of subprocesses of interest identified in the first stage. To implement the first substage, we propose a method that discovers a tree-based assembly model close to the original M²BOM and, therefore, well-suited to convey results to stakeholders from engineering. In doing so, we particularly focus on performance. Due to practical constraints, the actual process usually adheres to the provided M²BOM (e.g., parts cannot be missing and dependencies must be respected) and, thus, conformance checking tends to be less interesting.

Our main contributions are the investigation of so-called M²BOM-structured event logs and how multiple M²BOMs, with the help of domain knowledge and special types of material options, can be unified into a single common data model. We propose to detect bottlenecks based on this unified representation and to analyze the latter using a top-down approach. Finally, we illustrate the feasibility of our approach on an industrial-scale printer assembly use case provided by *Heidelberger Druckmaschinen AG*.

The remainder of this paper is structured as follows: Sects. 2 and 3 cover the related work and preliminaries, respectively. Section 4 presents the analysis approach, in particular, the discovery of M²BOM-based models in Sect. 4.2. We evaluate the approach in Sect. 5 and conclude our work in Sect. 6.

2 Related Work

There are many papers on improving the performance of manufacturing processes—for example, based on the principles of lean management [11]. We, however, focus on the more recent approach of using process mining to analyze and improve operational assembly processes. For a more detailed review on process mining for assembly-related processes (e.g., procurement), we refer the reader to [5]. One of the first case studies, conducted by Rozinat et al. [12], investigates the testing procedure of wafer scanners in terms of idle times and repeated tests. In this work, little additional structural information has been exploited. More similar to our use case in terms of independently manufactured parts is

the ship manufacturing process in [9], where multiple ship blocks are manufactured simultaneously. In contrast to our work, this work focuses on individual blocks only, applying trace clustering to identify similar intra-block assembly work flows. A comparison between planned and de facto schedules in block-level ship manufacturing processes can be found in [10]. Besides, model discovery for a coffee machine manufacturing process using standard automatic mining approaches has been investigated in [3]. Recently, Uysal et al. [15] analyzed the performance and evolution of an automotive production line. While this work also focuses on the performance of an assembly line, we consider more complex production lines and do not require a ground truth production model. However, both use cases share a similar assembly structure in common (i.e., major assembly steps, linked by additional structural information, and a set of unstructured assembly activities related to each major assembly step). More recently, Lorenz et al. [8] analyzed deviations between de jure and de facto models in sanitary product manufacturing using conformance checking. They emphasize that the major advantage of process mining over traditional methods is its adaptability to dynamic processes and that it can comprehensively consider entire cases. This strength is further underpinned by its application to production change point detection in [4]. Finally, a first framework for the end-to-end analysis of production processes using process mining has been proposed in [14].

3 Preliminaries

Throughout this paper, we use *out-trees* to model a bill of materials. Given a set of vertices V , a directed acyclic and weakly connected graph $T = (V, E)$ with $E \subseteq V \times V$ is a *tree*. We denote the set of vertices by $T_v = V$. A *rooted tree* is a tree with designated root vertex ρ_T and an *out-tree* is a tree where each edge points away from ρ_T . An *s-t path* for $s, t \in V$ is a sequence of edges $\langle e_1 = (s, v_1), e_2 = (v_1, v_2), \dots, e_k = (v_{k-1}, t) \rangle$, $e_i \in E, i = 1, \dots, k$.

In this work, we use restricted, loop-free, process trees to describe execution/replay semantics.

Definition 1 (Loop-free Process Tree). *Let \mathcal{A} denote a universe of activity labels such that $\tau \notin \mathcal{A}$. Let $\oplus = \{\rightarrow, \times, \wedge, \vee\}$ be the set of tree operators. A loop-free process tree is defined by the following production rules:*

- $a \in \mathcal{A} \cup \{\tau\}$ is a loop-free process tree
- $\bullet(T_1, \dots, T_n)$ for process trees $T_i, i = 1, \dots, n, n \geq 1$, and $\bullet \in \oplus$ is a loop-free process tree

Besides, given a process tree PT, we assume standard operator semantics for the defined language $\mathcal{L}(\text{PT})$ (compare [7]). Furthermore, to model the operational event data, we introduce the following universes and event projections.

Definition 2 (Event Universes). *To model the manufacturing event data, we define the universes of event identifiers, \mathbb{U}_{eid} ; product identifiers, \mathbb{U}_{pid} ; manufacturing activities, \mathbb{U}_a ; timestamps, \mathbb{U}_{time} ; material types, \mathbb{U}_{mtyp} ; material identifiers, \mathbb{U}_{mid} ; material id to material type mappings, $\text{mat} \in \mathbb{U}_{mid} \rightarrow \mathbb{U}_{mtyp}$; and events, $\mathcal{E} = \mathbb{U}_{eid} \times \mathbb{U}_{pid} \times \mathbb{U}_a \times \mathbb{U}_{time} \times \mathbb{U}_{mid}$.*

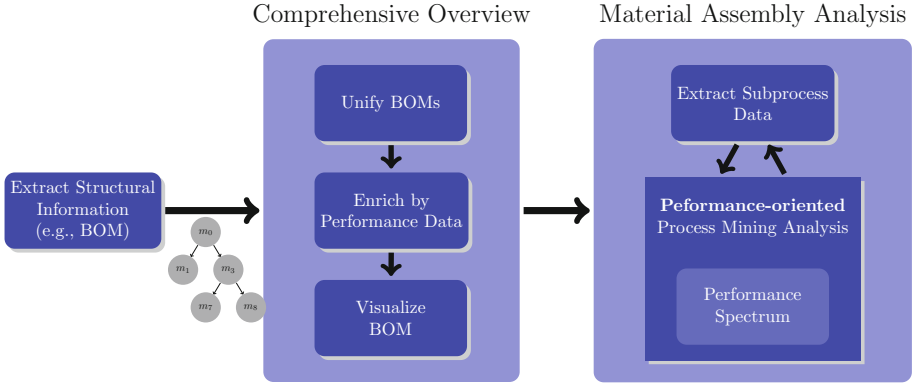


Fig. 1. Analysis methodology for M²BOM-structured event logs.

Notice that each event is related to a *single* material following the bill of materials-inspired idea that tasks can be attributed to a specific material where the assembly of multiple materials is attributed to the created new material. Given an event $e = (e_{id}, p_{id}, a, t, m_{id}) \in \mathcal{E}$, we denote the projection on the event id, product id, activity, timestamp, and material id by $\pi_{e_{id}}(e) = e_{id}$, $\pi_{p_{id}}(e) = p_{id}$, $\pi_a(e) = a$, $\pi_{time}(e) = t$, and $\pi_{m_{id}}(e) = m_{id}$, respectively. Furthermore, in a slight abuse of notation, we generalize the projection to sets, yielding multisets of attribute values, for example, $\pi_a(E) = [\pi_a(e) \mid e \in E]$ for $E \subseteq \mathcal{E}$. Finally, we introduce the following standard definition of an event log with the additional requirement that materials are not shared among different products.

Definition 3 (Assembly Event Log). An assembly event log (E, \leq_E) is a finite tuple of events $E \subseteq \mathcal{E}$ endowed with an ordering relation $\leq_E \subseteq E \times E$ such that: (i) \leq_E is a partial order, (ii) event identifiers are unique, i.e., $\forall e_1 \forall e_2 : \pi_{e_{id}}(e_1) = \pi_{e_{id}}(e_2) \Rightarrow e_1 = e_2$, (iii) ordering respects time, i.e., $\forall e_1 \forall e_2 \in E : e_1 \leq_E e_2 \Rightarrow \pi_{time}(e_1) \leq \pi_{time}(e_2)$, and (iv) no materials are shared, i.e., $\forall e_1 \forall e_2 \in E : \pi_{m_{id}}(e_1) = \pi_{m_{id}}(e_2) \Rightarrow \pi_{p_{id}}(e_1) = \pi_{p_{id}}(e_2)$.

4 Methods

In this section, we propose a top-down methodology for analyzing operational processes providing additional structural information and illustrate how a multi-level manufacturing bill of materials (M²BOM) can be exploited.

4.1 Analysis Methodology

A major challenge when analyzing operational processes is the potentially large number of assembly activities and materials. Additionally, particularly in special-purpose machine manufacturing, the number of orders is usually small. This generally negatively affects automatic process discovery techniques, yielding huge

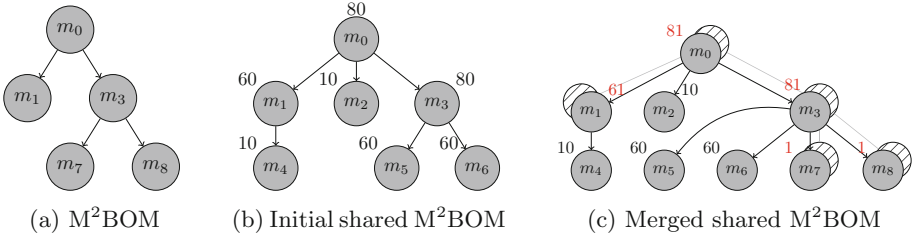


Fig. 2. Example of iteratively merging M²BOMs into a shared M²BOM.

and incomprehensible models. However, for humans, even with little domain knowledge, these processes are clearly structured and a lot of effort went into planning. In doing so, one material dependency modeling approach is by means of M²BOM. In the proposed process analysis methodology, depicted in Fig. 1, we therefore exploit this additional structural information to be able to visualize the process as a whole. To this end, we first extract the structural information. Then, a tree-based performance-aware overview over the entire process where vertices correspond to the materials, is created to show bottlenecks and to identify other points of interest (e.g., similar materials with relatively large performance differences). After identifying points of interest, particularly performance bottlenecks, a refined analysis of the associated subprocesses is conducted. Usually, manufacturing subprocesses are designed to be independent and, thus, little information is lost by focusing on a specific subprocess. Besides, the assembly of a specific material is often fairly sequential thereby facilitating the analysis. In doing so, a control-flow and conformance analysis tends to be less interesting; instead, the major focus must be on the performance. Given the reduced complexity of the subprocess, the performance spectrum [2], with time relative to timestamp of case start, is a well-suited tool because it allows for a high-resolution performance analysis. Finally, the subprocess analysis can be iterated drilling down further.

4.2 M²BOM-Structured Assembly Processes

A common approach to structure assembly processes is by means of multi-level manufacturing bills of materials (M²BOMs). Unfortunately, M²BOMs cannot be directly used to visualize the performance of multiple cases because products are often configurable and therefore have different bills of materials. Even though these configurations might be modeled for the customer in the ordering system, this information is lost when creating the actual manufacturing bill of materials. Thus, to provide a comprehensive assembly overview, this section proposes a method to discover an option-aware M²BOM from the data. To this end, we first merge a collection of M²BOMs into a common representation (compare Fig. 2) and then extend it into a proper configuration model (compare Fig. 3).

Conceptually, a M²BOM can be modeled by a tree as follows:

Definition 4 (Multi-level Manufacturing Bill of Materials (M²BOM)). Given a finite list of materials $M \subseteq \mathbb{U}_{m_{id}}$, a multi-level manufacturing bill of materials is an out-tree (M, D) .

An example M²BOM is depicted in Fig. 2(a). Since our approach operates on material types, we label the vertices by their type in our illustrations. In addition, to relate the event data of a particular product to its M²BOM, we define $\sigma_{m_{id}}(p, E) = \{\pi_{m_{id}}(e) \mid e \in E, \pi_{p_{id}}(e) = p_{id}\}$, which selects the materials used in the assembly of the product p_{id} for the assembly event log (E, \leq_E) . Next, we combine the classical event log and M²BOM into a *M²BOM-structured event log*.

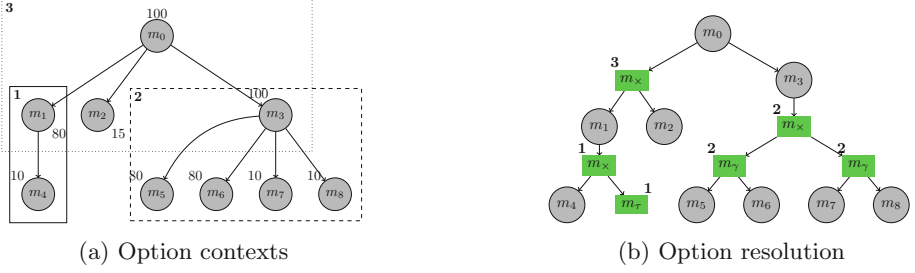
Definition 5 (M²BOM-structured event log). Let (E, \leq_E) denote an assembly event log. Let $BOM_E : \mathbb{U}_{p_{id}} \rightarrow (\mathcal{P}(\mathbb{U}_{m_{id}}) \times (\mathcal{P}(\mathbb{U}_{m_{id}} \times \mathbb{U}_{m_{id}})))$, $p_{id} \mapsto (M, D)$ be a function that assigns M²BOM to each product such that $M = \sigma_{m_{id}}(p, E)$. An M²BOM-structured event log is a tuple $((E, \leq_E), BOM_E)$.

We deliberately keep the event log and M²BOMs separate—requiring that material from the log occurs in the M²BOM and vice versa—to facilitate the use of other process mining techniques. Even though the performance of the assembly for a single product can be measured and projected onto the corresponding M²BOM using the M²BOM-structured event log, this does not provide aggregated statistics. Therefore, we first merge the M²BOMs into one shared M²BOM.

Definition 6 (Shared M²BOM). Let $E_{BOM} = ((E, \leq_E), BOM_E)$ be an M²BOM-structured event log. Let $M^\sigma \subseteq \mathbb{U}_{m_{id}} \times \mathbb{U}_{m_{typ}} \times \mathcal{P}(\mathbb{U}_{p_{id}})$ be a vertex set with id, material type, and product set projections $\pi_{m_{id}}(v) = m_{id}$, $\pi_{m_{typ}}(v) = m_{typ}$, and $\pi_{p_{id}}(v) = s_{p_{id}}$ for $v = (m_{id}, m_{typ}, s_{p_{id}}) \in M^\sigma$. $B^\sigma = (M^\sigma, D^\sigma)$ with $D^\sigma \subseteq M^\sigma \times M^\sigma$ is a shared M²BOM iff:

- B^σ is an out-tree (compare Definition 4)
- B^σ contains exactly the bills of materials present in E_{BOM} :
 - Each M²BOM is contained: for every product id $p_{id} \in \pi_{p_{id}}(E)$ there exists an injective homomorphism $h_{p_{id}} : (BOM_E(p_{id}))_v \rightarrow M^\sigma$ between B and B^σ that respects the material types and product id sets, i.e., $\forall m \in (BOM_E(p_{id}))_v$ ($\text{mat}(m) = \pi_{m_{typ}}(h(m)) \wedge p_{id} \in \pi_{p_{id}}(h(m))$).
 - B^σ contains only M²BOMs from the event log, i.e., $\forall v \in M^\sigma$ ($\{p_{id} \mid p_{id} \in \pi_{p_{id}}(E), h_{p_{id}}^{-1}(v) \neq \emptyset\} = \pi_{p_{id}}(v)$).

In the shared M²BOM, every vertex has an id (guaranteeing uniqueness), a type, and a set of products containing this material. Since M²BOM allows multiple materials instances having the same type, corresponding vertices between M²BOM and the shared M²BOM must be consistent in the type and location within the tree. We enforce this by the injective—no two material instances are mapped to the same vertex—homomorphism $h_{p_{id}}$. It ensures that every M²BOM can be type-consistently embedded into the shared M²BOM and that a vertex contains a product if and only if one of its materials is mapped to this vertex.



(a) Option contexts

(b) Option resolution

Fig. 3. Resolving material count mismatches by introducing options. (a) The option contexts and the order of retrieval. (b) The applied resolutions.

While the declarative definition does not provide a recipe for constructing the shared M²BOM, there is a straightforward iterative approach that merges vertices v, v' of trees T, T' if their and their ancestors' types are consistent (i.e., the material types on the $\rho_T - v$ and $\rho_{T'} - v'$ paths coincide). An example is depicted in Fig. 2, which, for simplicity, shows the cardinality of the product id sets instead of the actual sets. Furthermore, Fig. 2(c) also shows the homomorphism between the M²BOM (Fig. 2(a)) and the initial shared M²BOM (Fig. 2(b)).

While the shared M²BOM allows for a visualization of aggregated projected statistics, it cannot properly capture material frequency differences in terms of certain materials being optional or choices between materials. Besides, it is also desirable to link the shared M²BOM to a proper process model to be able to apply other process mining techniques. For example, process simulation can be used for production planning. To this end, we transform the shared M²BOM into an option M²BOM that, in turn, can be directly related to a process tree. The option M²BOM models optional materials and choices using dedicated special material types $m_\gamma, m_\vee, m_\times,$ and m_τ . While m_γ is used to create material groups; $m_\vee, m_\times,$ and m_τ directly correspond to their pendants in process trees. An example of the transformation is depicted in Fig. 3(b), showing that, for example, a customer may choose between m_1 and m_2 .

Definition 7 (Option M²BOM). Let $\mathbb{U}_{m_{typ}}^o = \mathbb{U}_{m_{typ}} \dot{\cup} \{m_\gamma, m_\times, m_\vee, m_\tau\}$ denote an extended material type universe. An option M²BOM is an out-tree $B^o = (M^o, D^o)$ with $M^o \subseteq \mathbb{U}_{m_{typ}}^o \times \mathbb{U}_{m_{id}}, D^o \subseteq M^o \times M^o$ such that m_τ only occurs as leaf vertex adjacent to a choice vertex of type m_\times or m_\vee .

An option M²BOM directly corresponds to a process tree where material groups are modeled by concurrency and non-leaf materials as sequences of concurrent child material manufacturing followed by the assembly of the parents themselves. Accordingly, we define the process tree of an option M²BOM as follows:

Definition 8 (Process Tree of an Option M²BOM). Given an option M²BOM $B^o = (M^o, D^o)$ and a vertex $v \in M^o$, the process tree $\text{PT}_{B^o}(v)$, rooted

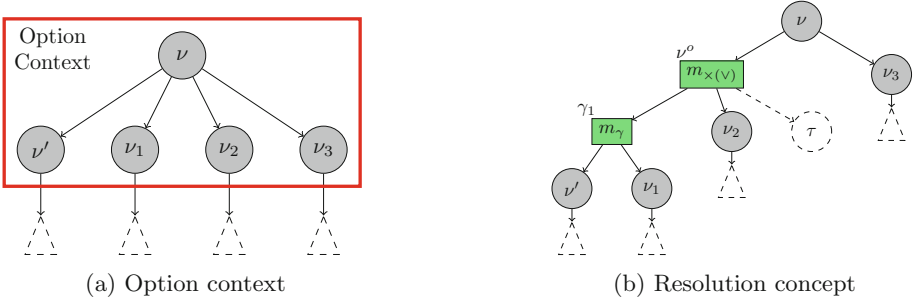


Fig. 4. Transforming a shared M²BOM into an option M²BOM by (a) identifying option contexts and (b) applying different resolution strategies.

at v , of the option M²BOM is recursively defined as follows:

$$PT_{B^\circ}(v) = \begin{cases} \times (PT_{B^\circ}(c_1), \dots, PT_{B^\circ}(c_n)) & \text{if } v = m_\times(v) \\ \wedge (PT_{B^\circ}(c_1), \dots, PT_{B^\circ}(c_n)) & \text{if } v = m_\gamma \\ \pi_{m_{typ}}(v) & \text{if } v \text{ is a leaf} \\ \rightarrow (\wedge (PT_{B^\circ}(c_1), \dots, PT_{B^\circ}(c_n)), \pi_{m_{id}}(v)) & \text{if } v \in \mathbb{U}_{m_{typ}} \end{cases} \quad (1)$$

where (c_1, \dots, c_n) is an arbitrary enumeration of the children of v . We denote the process tree obtained for the root of B° by PT_{B° (i.e., $PT_{B^\circ} := PT_{B^\circ}(\rho_{B^\circ})$).

To relate an option M²BOM to a concrete M²BOM, we, first, introduce the material reduction of an option M²BOM B° that reduces B° to an M²BOM. It is obtained by repeatedly replacing edges $(s, u), (u, t)$ with $s, t \in \mathbb{U}_{m_{typ}}, u \in \{m_\gamma, m_\times, m_\nu\}$ by (s, t) and removing m_τ leaves and vertices without adjacent edges. We denote the material reduction by $B_{|\mathbb{U}_{m_{typ}}}^\circ$. For example, Fig. 4(a) shows the material reduction of the resolution depicted in Fig. 4(b). Using the material reduction and Definition 7, we can establish the link between M²BOM-structured event logs and option M²BOMs. M²BOM B is compatible with an option M²BOM if for each material in B there is a corresponding material in $B_{|\mathbb{U}_{m_{typ}}}^\circ$ and if B is a valid combination of materials w.r.t. the options modeled in B° (e.g., no mandatory material is missing or exclusive material options are respected). To this end, we require that a potentially valid production plan of B (i.e., materials are ordered such that child materials are manufactured before their parent materials), is contained in the language of the process tree PT_{B° .

Definition 9 (Option M²BOM Compatibility). Given M²BOM $B = (M, D)$ and an option M²BOM $B^\circ = (M^\circ, D^\circ)$ with its material reduction $B_{|\mathbb{U}_{m_{typ}}}^\circ = (M_r^\circ, D_r^\circ)$, B realizes B° if there exists an injective homomorphism $h: M \rightarrow M_r^\circ$ between B° and $B_{|\mathbb{U}_{m_{typ}}}^\circ$ satisfying the following conditions: (i) material types are respected, i.e., $\forall m \in M \pi_{m_{typ}}(m) = \pi_{m_{typ}}(h(m))$ and (ii) the post-order traversal $\langle v_1, \dots, v_n \rangle$ of vertices in B , $\langle h(v_1), \dots, h(v_n) \rangle$ is in the language of the process tree of B° , i.e., $\langle h(v_1), \dots, h(v_n) \rangle \in \mathcal{L}(PT_{B^\circ})$.

Finally, a BOM-structured event log is compatible with an option M²BOM if M²BOM of every product is compatible. Similar to the construction of the shared M²BOM, there is a straightforward approach based on comparing product sets to construct a compatible option M²BOM from the shared M²BOM of a BOM-structured event log. Figure 4 illustrates the major steps; first, option contexts induced by product count mismatches between parent and child vertices are iteratively retrieved in bottom-up order. Given a count mismatch between the products associated with parent v and child v' (i.e., $|\pi_{pid}(v)| \neq |\pi_{pid}(v')|$), the option context comprises v and *all* its children as these might be in a, so far undiscovered, choice relation with v' . This mismatch can then be resolved by introducing group, exclusive choice, or non-exclusive choice nodes as well as the possibility to skip certain materials; the concept of the resolution is depicted in Fig. 4(b). The resolution usually requires domain knowledge because the data may not contain all valid configurations. For example, two configurations might, by incident, never occur together even though they could. Therefore, making both optional should be preferred over an exclusive choice. Finally, the set of product ids covered by a newly introduced vertex is equal to the union of its successor’s cover. Notice that for optional subtrees, the counting argument above has to be slightly modified so that these vertices are not handled repeatedly. Eventually, we obtain a valid option M²BOM after resolving all options. Figure 3(b) shows a complete option resolution, including the order of steps. First, m_4 is found to be an optional part of m_2 . Next, we discover an occurrence mismatch between m_3 and its child materials that can be resolved by a choice between two material groups. Finally, an exclusive choice between m_1 and m_2 is introduced.

5 Case Study

We evaluated the proposed methodology using real-world data from *Heidelberger Druckmaschinen AG*—a global manufacturer of offset, digital, and label printing presses. The company not only offers special-purpose machines but also provides services for the entire industrial printing value chain. The event data comprises events for several of hundred offset printers of different models and configurations. In agreement with the company’s confidentiality policy, we anonymized the data (i.e., the activities, materials, and time spans). However, to give a high-level intuition, we assigned the materials in the first four M²BOM levels expressive names. As depicted in Fig. 5, the root element is the printer, the second level comprises logistics materials, the third level’s material is required to finalize the machine (*Final Comp.*), and the fourth level comprises the major large components of an offset printer (*Large Comp.*). In addition, each event contains a reference to its and its parent’s material id, which was used for the automated M²BOM construction.

BOM-Based Overview. In coordination with the stakeholders, we applied the option M²BOM discovery approach to the most frequently sold product. We obtained an option M²BOM containing more than 250 different materials and

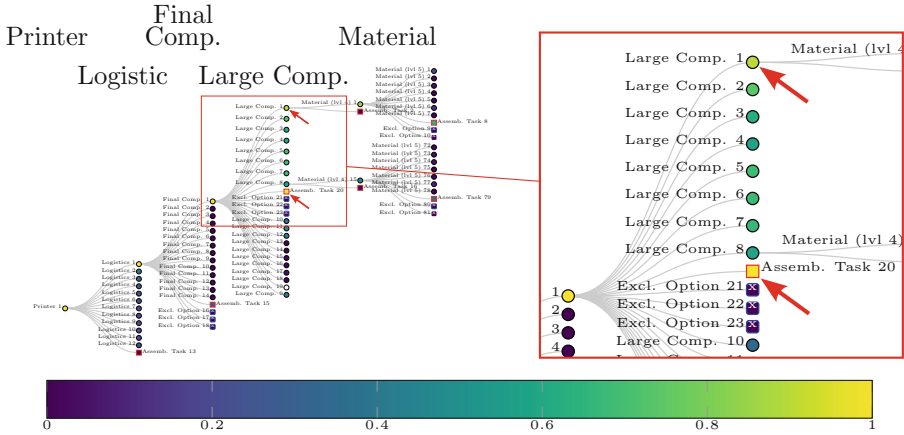


Fig. 5. An excerpt of our visualization of the option M²BOM, discovered for the most frequently sold printer model (anonimized time scale).

approximately 25 choices. For each vertex v , we computed the median assembly time, i.e., the timespan between the start and complete timestamp of the first and the last event related to a material in the subtree rooted at v . Moreover, we included the business hours and the factory calendar in the computations. An excerpt of the resulting option M²BOM colored by the median assembly times is depicted in Fig. 5. Starting at the root node, we expanded each level’s most performance-relevant material up to a depth of three. Each circular vertex corresponds to a material, while squares correspond to options or a special *activity material* that subsumes all assembly tasks related to the parent vertex. For example, *Assembly Task 20* subsumes the events required to assemble *Final Comp. 1* using the materials on the fourth level. Besides, *Excl. Option 21* shows an optional printer part. Considering the performance of the assembly, this visualization clearly shows the most time-consuming operations—namely, *Assembly Task 20* and *Large Comp. 1*. In contrast to a plain list of assembly times, Fig. 5 also depicts the relations between the materials, facilitating performance comparison. Knowing that *Large Comp. 1-8* are similar materials, Fig. 5 shows median assembly time differences between these components. In particular, the increased assembly duration of *Large Comp. 1* compared to *Large Comp. 8* is due to a slightly increased complexity of the respective assembly tasks. However, we will focus on *Assembly Task 20*, the most time-consuming step.

Bottleneck Analysis. Next, we investigated the major bottleneck, *Assembly Task 20*. To this end, we extracted the corresponding events for all printers of the considered model and discovered a process model using the default Inductive Miner infrequent [7] algorithm. As expected, the resulting model was mostly sequential and exhibits only little concurrency. Using this model, we created the token flow-based *performance spectrum* [2]. In doing so, we exploited additional

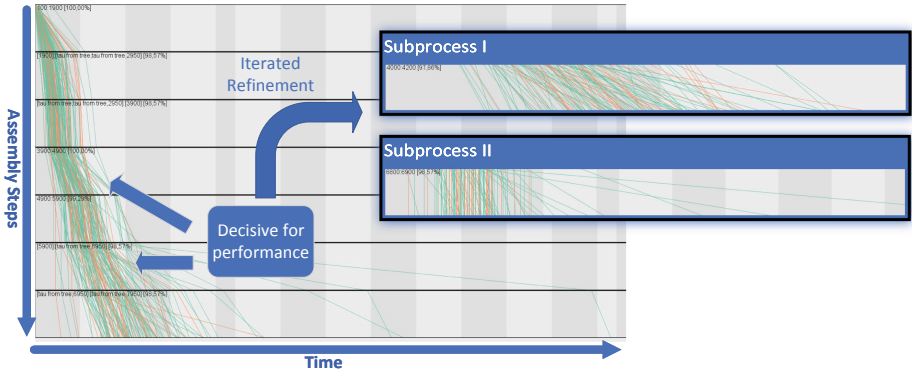


Fig. 6. Performance Spectrum for the most critical assembly activity. (Color figure online)

domain knowledge to identify sections in the subprocess. The resulting performance spectrum is depicted at the left hand side of Fig. 6; the vertical axis shows the flow of cases through the identified sections, while the horizontal axis shows the time relative to the start of the assembly. We further differentiate between standard machines (orange) and machines with additional customization and features (cyan). Using the performance spectrum, we identified two crucial sections, where times differ significantly among various machines. By iterating the subprocess analysis step, we were finally able to identify the decisive assembly tasks in terms of overall performance within the two sections. The performance spectra for these activities are depicted on the right hand side of Fig. 6.

6 Conclusion

In this work, we propose an analysis methodology for conducting a process mining analysis in operational (assembly) processes that provide additional structural information in terms of multi-level manufacturing bills of materials. Our analysis methodology uses a top-down approach that first creates an overview over the entire process, exploiting the available additional structural information, and then analyzes subprocesses in more detail. In particular, we propose an option BOM-based visualization and provide a method to discover an option M²BOM from the assembly event data. We demonstrate the applicability of the analysis methodology, particularly the discovery and visualization of the option M²BOM, on a real-world industrial-scale printer manufacturing use case.

For future work, we plan to extend the option M²BOM mining approach to incorporate different printer models and to apply it to additional manufacturing domains. Moreover, incorporating process variant comparison approaches, particularly w.r.t. performance, would be interesting. Finally, as even subprocesses can be quite large, we aim to investigate methods that automatically detect performance-critical parts in performance spectra.

Acknowledgements. Funded by the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) under Germany's Excellence Strategy—EXC-2023 Internet of Production—390621612. We thank the Alexander von Humboldt (AvH) Stiftung for supporting our research.

References

1. van der Aalst, W.M.P., Brockhoff, T., Ghahfarokhi, A.F., Pourbafrani, M., Uysal, M.S., van Zelst, S.J.: Removing operational friction using process mining: challenges provided by the internet of production (IoP). In: Hammoudi, S., Quix, C., Bernardino, J. (eds.) DATA 2020. CCIS, vol. 1446, pp. 1–31. Springer, Cham (2021). https://doi.org/10.1007/978-3-030-83014-4_1
2. van der Aalst, W.M.P., Tacke Genannt Unterberg, D., Denisov, V., Fahland, D.: Visualizing token flows using interactive performance spectra. In: Janicki, R., Sidorova, N., Chatain, T. (eds.) PETRI NETS 2020. LNCS, vol. 12152, pp. 369–380. Springer, Cham (2020). https://doi.org/10.1007/978-3-030-51831-8_18
3. Bettacchi, A., Polzonetti, A., Re, B.: Understanding production chain business process using process mining: a case study in the manufacturing scenario. In: Krogstie, J., Mouratidis, H., Su, J. (eds.) CAiSE 2016. LNBIP, vol. 249, pp. 193–203. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-39564-7_19
4. Chiò, E., Alfieri, A., Pastore, E.: Change-point visualization and variation analysis in a simple production line: a process mining application in manufacturing. *Procedia CIRP* **99**, 573–579 (2021)
5. Dreher, S., Reimann, P., Gröger, C.: Application fields and research gaps of process mining in manufacturing companies. In: LNI, pp. 621–634 (2021)
6. van Eck, M.L., Lu, X., Leemans, S.J.J., van der Aalst, W.M.P.: PM²: a process mining project methodology. In: Zdravkovic, J., Kirikova, M., Johannesson, P. (eds.) CAiSE 2015. LNCS, vol. 9097, pp. 297–313. Springer, Cham (2015). https://doi.org/10.1007/978-3-319-19069-3_19
7. Leemans, S.J.J.: Robust Process Mining with Guarantees. Ph.d. thesis, Eindhoven University of Technology (2017)
8. Lorenz, R., Senoner, J., Sihni, W., Netland, T.: Using process mining to improve productivity in make-to-stock manufacturing. *IJPR* **59**(16), 1–12 (2021)
9. Park, J., Lee, D., Zhu, J.: An integrated approach for ship block manufacturing process performance evaluation: case from a Korean shipbuilding company. *Int. J. Prod. Econ.* **156**, 214–222 (2014)
10. Park, M., Song, M., Baek, T.H., Son, S.Y., Ha, S.J., Cho, S.W.: Workload and delay analysis in manufacturing process using process mining. In: Bae, J., Suriadi, S., Wen, L. (eds.) AP-BPM 2015. LNBIP, vol. 219, pp. 138–151. Springer, Cham (2015). https://doi.org/10.1007/978-3-319-19509-4_11
11. Pfeiffer, W., Weiß, E.: Lean Management: Grundlagen der Führung und Organisation lernender Unternehmen. Erich Schmidt Verlag GmbH & Co KG (1994)
12. Rozinat, A., de Jong, I., Günther, C., van der Aalst, W.M.P.: Process mining applied to the test process of wafer steppers in ASML. *IEEE Trans. Syst. Man Cybern. Part C Appl. Rev.* **39**(4), 474–479 (2009)
13. SAP: Multilevel bill of material (BOM) (2021). <https://blogs.sap.com/2021/05/07/multilevel-bill-of-material-bom>

14. Schuh, G., Gützlaff, A., Schmitz, S., van der Aalst, W.M.P.: Data-based description of process performance in end-to-end order processing. *CIRP Ann.* **69**(1), 381–384 (2020)
15. Uysal, M.S., et al.: Process mining for production processes in the automotive industry. In: *BPM Industry Forum* (2020)

Open Access This chapter is licensed under the terms of the Creative Commons Attribution 4.0 International License (<http://creativecommons.org/licenses/by/4.0/>), which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons license and indicate if changes were made.

The images or other third party material in this chapter are included in the chapter's Creative Commons license, unless indicated otherwise in a credit line to the material. If material is not included in the chapter's Creative Commons license and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder.

