Ontology-based mapping approach for automatic work packaging in modular construction

Xiao Li[1], Chengke Wu[2], Fan Xue[3], Zhile Yang[4], Jinfeng Lou[5], Weisheng Lu[6]

This is the peer-reviewed post-print version of the paper:
Li, X., Wu, C., Xue, F., Yang, Z., Lou, J., & Lu, W. (2022). Ontology-based mapping approach for automatic work packaging in modular construction. *Automation in Construction*, 134, 104083. Doi: 10.1016/j.autcon.2021.104083

Abstract

Many cross-knowledge domain tasks requiring various professional backgrounds have been transferred from construction site to factory in modular construction (MC). Forming effective work packages in MC considering the complexity of product breakdown structure becomes crucial for task planning and execution. However, the definition of optimal work packages in MC is currently time-consuming and knowledge-inadequate. This study aims to develop a dynamic ontology-based mapping (DOM) model for generating semantic-enrich work packages. To this end, ontologies of module products, module topology, and tasks are first established. The customized Latent Dirichlet Allocation model and weighted K-means clustering method for mapping products to tasks and grouping tasks into work packages are then developed. Finally, an evaluation experiment of the proposed DOM model in a real MC case study demonstrated that it could improve the accuracy and efficiency of the dynamic work packaging process and pave the way for collaborative planning in MC.

Keywords: Work Package; Ontology; K-means; Latent Dirichlet Allocation; Modular Construction; Project Planning

[1] RGC Postdoctoral Fellow, Department of Real Estate and Construction, The University of Hong Kong, Hong Kong SAR. Email: xl1991@hku.hk, Tel.: (852) 5537 0842, ORCiD: 0000-0001-9702-4153;
[2] School of Design and Built Environment, Curtin University, Bently 6102, Western Australia, Australia. Email: chengke.wu@postgrad.curtin.edu.au, Tel.: (852) 5537 0842, ORCiD: 0000-0001-9702-4153; *: Corresponding author
[3] Assistant Professor, Department of Real Estate and Construction, The University of Hong Kong, Hong Kong SAR. Email: xuef@hku.hk, Tel.: (852) 3917 4174, ORCiD: 0000-0003-2217-3693
[4] Shenzhen Institue of Advanced Technology, Chinese Academy of Sciences, Shenzhen, China
[5] Ph.D. student, Department of Real Estate and Construction, The University of Hong Kong, Hong Kong SAR. Email: waseljf@connect.hku.hk, Tel.: (852) 9423 1962, ORCiD: 0000-0001-5748-0146
[6] Professor, Department of Real Estate and Construction, The University of Hong Kong, Hong Kong SAR. Email: wilsonlu@hku.hk, Tel.: (852) 3917 7981, ORCiD: 0000-0003-4674-0357

1. Introduction

Modular construction (MC) is an innovative construction method to manufacture the facility (e.g., infrastructure, building) products in the factory and deliver them to the site for assembly (Gann, 1996; Teng et al., 2017; Ekanayake et al., 2020). MC has the potentials to reach Construction 4.0 since it has been recognized with compelling advantages over traditional cast-in-situ construction, such as shortened construction times (Eriksson et al., 2014), ensured quality (Goh and Goh, 2019), reduced site labor (Xu et al., 2020), and better working environment (Hammad et al., 2019). As MC involves multi-specialty and cross-domain knowledge in producing a facility, factories' production still requires assigning tasks to various subcontractors, specialized work teams, or even robotics and automatic machinery (Zhu et al., 2021; Arashpour et al., 2018). For example, a two-tower hostel residential project needs more than 600 workers at the peak of production, and a single housing module requires a doze of trades to work on various systems, such as structure, door/window, wall, wet, print, mechanical, electrical and plumbing (MEP) equipment (Lawson et al., 2014). These physically connected systems in a module are manufactured by coordinating interdependent tasks, which need seamless project interface planning (Salama et al., 2017).

Work Breakdown Structure (WBS) is an efficient project planning tool that was jointly developed by the U.S. Department of Defense (DoD), NASA and the U.S. aerospace industry in 1962 (DoD and NASA, 1962). It has now been widely extended for construction project management and planning (Siami-Irdemoosa et al., 2015; Li and Lu, 2017; Jung and Woo, 2004; Sutrisna et al., 2018). A WBS is a hierarchical decomposition of the total work scope in a project, and a *work package* is the smallest element in WBS for planning one or more *executable tasks* (Project Management Institute, 2021). The responsibility for executing a work package is normally assigned to a single person or organizational unit. The benefits of using work packages for MC project planning have also been recognized (Isaac et al., 2017; Li et al., 2019a, 2019b; Liu et al., 2015, 2016; Wu et al., 2021b): (i) it offers the fragmented MC project team members with clear instructions of their roles and responsibilities; (ii) it allows concurrent tasks to be simultaneously executed on an MC project; (iii) it helps measure the performance of schedule and cost by using techniques, such as earned value methods; (iv) it also supports risks, constraints, and disturbances management at a task level.

The definition and formation of work packages are the very first step before conducting efficient project planning. As work packages are mainly formed through manually decomposing the WBS based on experience and knowledge from project managers in current practice, it is inefficient and prone to omit critical tasks (Ramasesh et al., 2014). To address this issue, Ibrahim et al. (2009) automatically generated the work packages using predefined

attributes of BIM components. Isaac et al. (2017) further considered the topological relations, sequences, and interfaces between specific BIM components to form more effective work packages. Despite previous studies bridging object-oriented BIM components with the tasks-oriented work packages, forming work packages in product-oriented MC, challenges still exist. For example, (1) mapping rules between MC products and tasks are manually defined rather than modeled automatically; (2) The products content (e.g., bill of material) and task contents in work packages may vary with task execution approaching. Thus, there is a lack of an efficient approach to automatically form semantic-rich work packages by dynamically mapping products with tasks and then enriching the semantics of tasks for easing the packaging process.

This study aims to develop a dynamic ontology-based mapping (DOM) model for generating semantic-enrich work packages. It has three concrete objectives: (1) to explore and establish the ontologies of products, topology, and tasks; (2) to investigate the mapping models among products, tasks, and work packages; (3) to develop work packages generation method and validate them in an MC case study. The rest of the paper is organized as follows. After this introductory section is Section 2, which elaborates on related studies in WBS and work package formation, ontology modeling and clustering techniques. Section 3 delineates the research method, including the ontology models for products, topology, tasks, and the customized mapping model of Latent Dirichlet Allocation and improved work packaging method of weighted K-means clustering. Section 4 states the experimental evaluation in a real MC case study to demonstrate the results of automatic work package formation processes. Discussions are conducted in Section 5, and conclusions are drawn in Section 6.

2. Literature review

This study mainly involves four relevant topics: WBS generation, work package formation, ontology modeling, and data mining and clustering techniques. The review of these four topics can help provide basics for work packaging, identify limitations of current work packaging methods, investigate the possibility of using ontology to model the knowledge of products, tasks, and their relations to facilitate work packaging, and justify the methods used for product-task mapping and tasks clustering in this study. The details have been summarized below:

The first type of relevant study is to generate WBS in project management. WBS has been widely used to hierarchically decompose a project into manageable pieces (e.g., work package) for reducing project complexity. Golpayegani and Emamizadeh (2007) used neural networks to recognize the components and relationships in the project WBS. Siami-irdemoosa et al. (2015) then applied a similar method to generate WBS for the complex underground construction project. Torkanfar and Azar (2020) further developed a similarity measurement to conduct a semantic comparison of WBSs for achieving knowledge reuse. To

improve the dynamic WBS generation, Lee et al. (2010) proposed a system to support bi-directional transformation between processes and WBS by using the design structure matrix. Park and Cai (2017) also established an automated linking mechanism between tasks and BIM objects to help generate a dynamic BIM database. In modular construction, as the incompatibility between product-oriented off-site manufacturing and activity-oriented on-site construction can reduce the seamless interface and integration, Sutrisna et al. (2018) proposed a hybrid WBS-matrix to bridge the off-site PBS and on-site WBS, and each prefabricated module is defined as a work package. Many efforts have been made to generate WBS in MC or project management. However, previous works mainly focusing on forming the static and dynamic structure of WBS rather than dynamically defining the smallest and executable units, such as work packages, which is the manageable connection between planning and execution.

The second type of relevant study is to define effective work packages in project management. It is quite challenging because it requires decoupling and coupling the dependencies between tasks. For example, Raz and Globerson (1998) defined the work packages by considering the factors of cost and schedule estimation, progress control, network construction, internal cohesion, cash flow, and risk management. Abuwarda and Hegazy (2016) determined the work packages by selecting network paths and construction methods. As the project scope is hard to be fully decomposed through the above dynamic factors, recent studies utilized dynamic databases or BIM from the real project practices to help form the work packages. For example, Isaac et al. (2017) used BIM to identify topological relations between components and define construction sequences to generate work packages. Wang et al. (2020) generated the work package instances using the work package templates stored in the database and BIM. Using electrical documents, such as workflow, bill of materials, and production schedules, is common practice for task planning and execution in modular construction. The topic modeling approach has been investigated to efficiently map two entities in documents. For example, Aversano et al. (2016) proposed an approach for evaluating the alignment level between a business process and the supporting software systems. Marcus and Maletic (2003) used latent semantic indexing to automatically identify traceability links between system documentation and program source code. Pessiot et al. (2010) then extended the Probabilistic Latent Semantic Analysis (PLSA) model for document clustering. Al-Anazi et al. (2016) compared three clustering techniques: K-means, K-means fast, and K-medoids in document clustering using measures of cosine similarity, Jaccard similarity, and correlation coefficient. Baskara et al. (2016) used LDA to discover a traceability link between business processes and software components. Currently, mapping rules between BIM objects and work packages are manually defined rather than modeled automatically. Particularly for product-oriented MC with massive components, material and related tasks, there is a lack of an automatic mapping model to transform the MC products to

work packages.

The third type of relevant study is to develop ontology models in the construction field. Ontology is a graphical approach to map the knowledge domain using nodes and relations (Zhou et al. 2016). Studies related to ontology in construction could be divided into three groups: (1) information extraction, (2) knowledge modeling, (3) reasoning and conformance check. For example, Liu et al. (2016) developed an ontology-based semantic approach to extract quantity take-off information for workface planning. Zhang et al. (2015) modeled safety knowledge with an ontology model by linking tasks, methods, and job hazards. Wu et al. (2021a) used reasoning rules to improve traditional ontology models for constraints management. To enrich a detailed WBS for progress monitoring, Han et al. (2015) established an ontology model for construction sequence rationale, including physical relationships among components, path interference, code regulations, trade interaction. However, there is lacking ontology modeling for work packages in modular construction, particularly for mapping the relationships among products, topology and tasks.

The fourth type of relevant study is the mining and clustering of construction information. LDA is an unsupervised probabilistic model extensively applied to analyze discrete and unstructured data, such as texts. The LDA model first learns to identify main topics from a large archive of text documents (i.e., the training process). In this stage, the LDA model, in essence, clusters documents based on the topics. The number of topics can be pre-defined according to certain criteria, e.g., perplexity and similarity. The documents used for training the LDA model are also called a text corpus. After training, the LDA model can assign topics to a new document (i.e., the validation or testing process) (Blei et al., 2003). In the construction sector, LDA has been adopted by many studies for topic modeling in various aspects, including identifying main on-site issues and their changes over time (Lin et al., 2020), understanding the perceptions of governments towards mitigating environmental impacts of highway construction projects (Wu et al.,2019), investigating main types of lawsuit cases (Jallan et al., 2019), and categorizing main hazards from injury reports (Zhong et al., 2020). Although Dirichlet topic distribution cannot capture correlations among words (Hoffman et al., 2010), the components in MC do not require sequential correlations for component-task mapping. Meanwhile, the LDA model is highly modular and can be easily extended and embedded in more complicated models to improve the accuracy of component-task mapping. For example, LDA can be enriched with topology ontology models to analyze spatial relations among semantically related components and then cluster them based on certain categories.

3. Research method

As the bill of material (BoM) and workflow/schedule may change with the project's progress,

the proposed dynamic ontology-based mapping (DOM) model aims to facilitate generating semantic-enrich work packages, as shown in Figure 1, which includes the steps of ontology development, product-task mapping, and task-work package mapping.

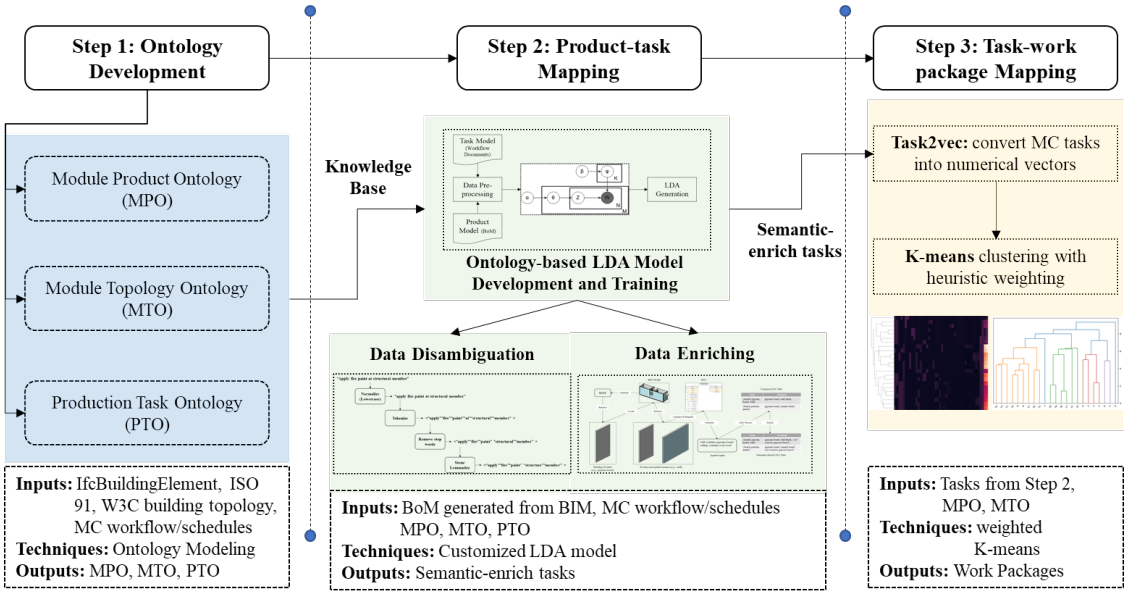


Figure 1 The proposed method: DOM model

3.1.Ontology development

3.1.1. Module product ontology (MPO)

This ontology (see Figure 2) includes two main branches to cover common classes of components and materials in MC projects under the top class 'Module Product'. The ontology has a maximum of four class levels, e.g., {Module Product, Component, Beam, T-Beam}. The 'Component' branch is built by referring to the IfcBuildingElement ontology of buildingSmart, modified according to MC projects' features. For instance, the 'Furniture' class is added, as installing furniture is a critical task during producing modules. The 'Material' branch is built based on material classification standards (International Organization for Standardization, 2021) for generating task2vec vectors (see Section 3.3). An 'isMadeBy' relation can be defined to relate classes in the two main branches. However, the MPO is not employed for information searching (e.g., searching for spatial relations introduced in Section 3.2.3). Thus, no semantic relations in this ontology are defined except the 'subclass' relation.

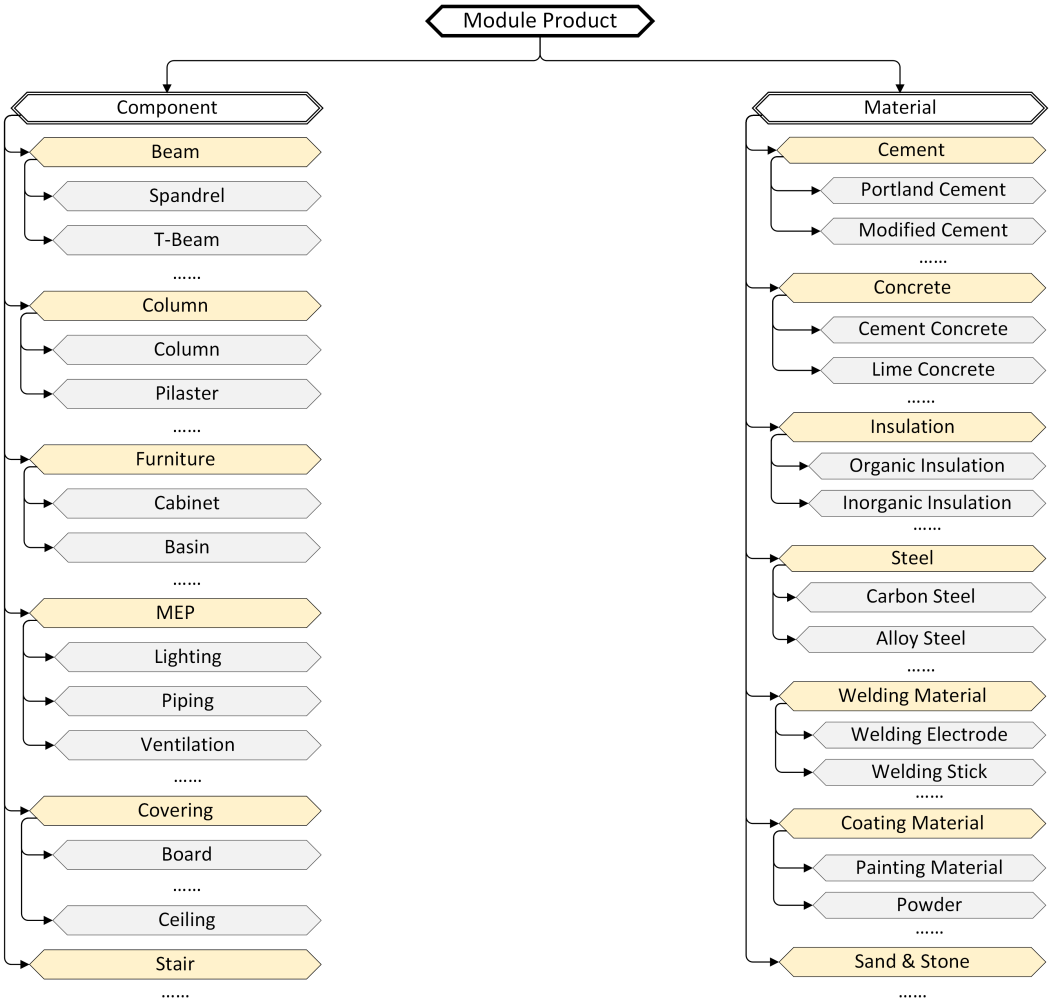


Figure 2 Conceptual structure of the MPO (not fully expanded)

3.1.2. Module topology ontology (MTO)

The MTO defines the spatial topology of a prefabricated module. The ontology released by the W3C, the minimal ontology for defining relationships among module products, is used to build MTO. Figure 3 shows classes and relations in this ontology. The MTO has two main classes, 'Zone' and 'Interface' and can be expanded to the fifth level. The 'Interface' class can qualify the relationships among module products or zones, e.g., a door between two rooms. The class has three sub-classes representing common interfaces, i.e., door, window, and MEP (mechanical, electrical, and plumbing) openings. Instances of the classes can be linked to instances of 'Zone' and MPO classes through the relation 'interface-of'. On the other hand, 'Zone' defines a spatial concept that can be extended in a maximum of three dimensions. 'Zone' has four sub-classes: 'Building', 'Story', 'Space', and 'Plane'. Their instances can be linked using three spatial relations, i.e., 'adjacent,' 'intersects', and 'contains'. The 'intersects' and 'adjacent' define the relationships between two zones that share an interface while do or do not intersect with each other, respectively. The 'contains' defines the

subsumption relationships among zones. A building often contains multiple stories, which can, in turn, contain one or more spaces horizontally connected. Space has bounded 3D spatial extent and provides certain functions (e.g., a toilet and kitchen). Finally, space contains planes that can be either vertical (e.g., a floor or ceiling) or horizontal (e.g., a wall).

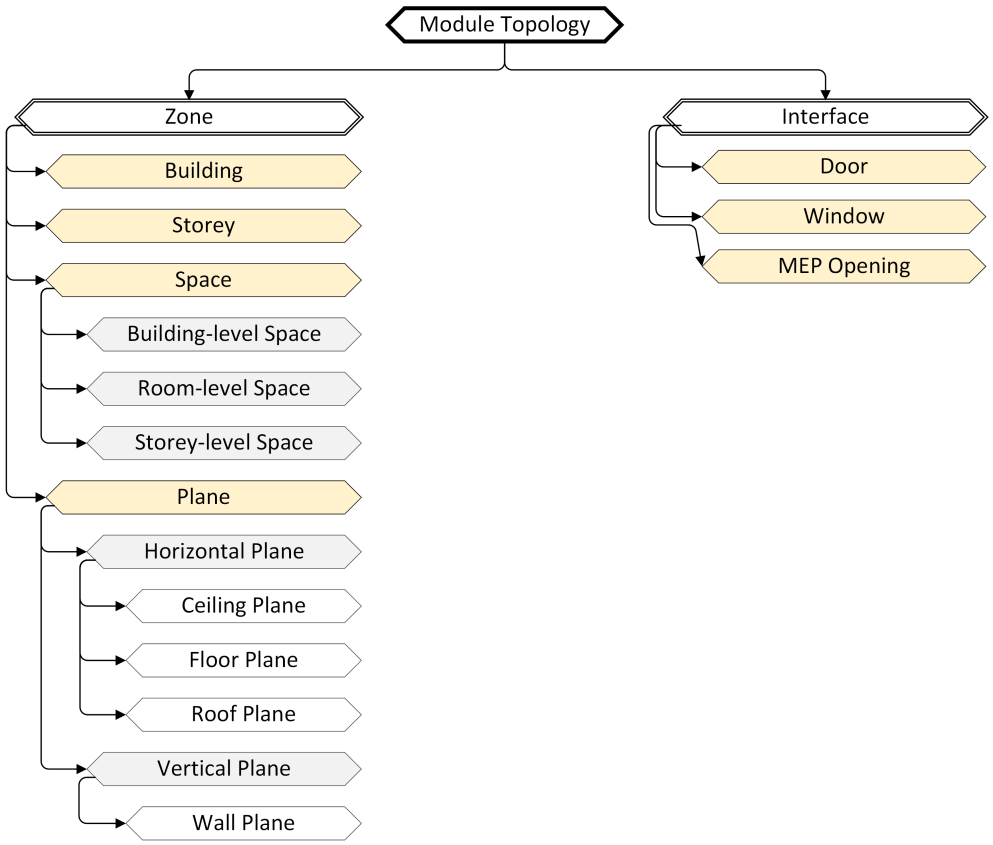


Figure 3 Conceptual structure of the MTO

3.1.3. Production task ontology (PTO)

This ontology defines essential tasks to fabricate a module. A module is typically made of many module products and can require different methods. The construction of the PTO is based on these distinct methods referring to the work in An et al. (2019). Figure 4 shows the conceptual structure of the PTO, which has a maximum of three levels. The PTO is built to form the lexicon for disambiguation introduced in Section 3.2.2. Hence, it also only considers one semantic relation, i.e., 'subclass'.

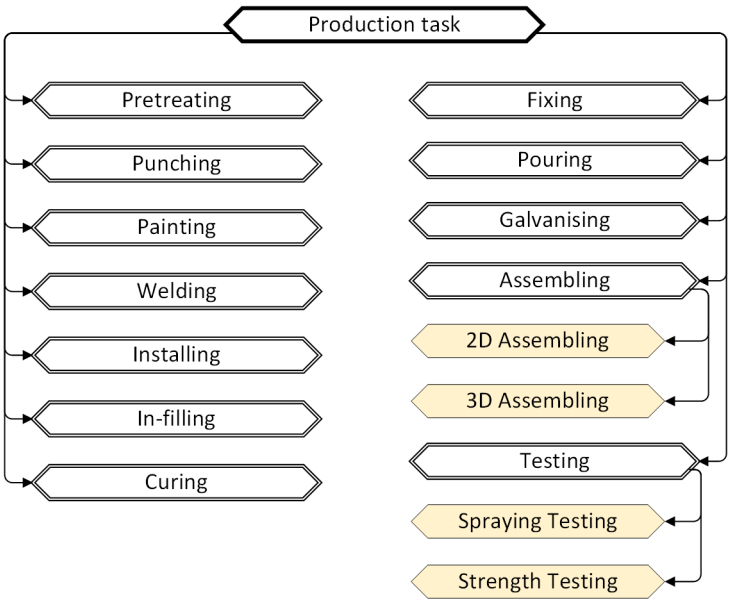


Figure 4 Conceptual structure of the PTO

3.2. Product-task mapping using LDA

3.2.1. LDA model development and training

As mentioned, the LDA model is originally used to identify topics of text documents. It assumes each document is described by topics, and each topic has a distribution over words. The model is trained by simulating the process of generating documents based on topics of documents and words. LDA relies on two Dirichlet distributions, i.e., **DD1** $P(\theta;\alpha)$ and **DD2** $(\phi;\beta)$, and two multinominal distributions, i.e., **MD1** $P(Z|\theta)$ and **MD2** $P(W|Z)$. The α and β are hyper-parameters, θ and ϕ are latent variables, and Z and W represent topics and words, respectively. For each of the ***M*** documents to be generated, the **DD1** determines the topic of the document, from which **MD1** is derived to determine the topic of each word in the document. The **MD1** is called ***N*** times (***N*** is the number of words, i.e., the document length). In this stage, each word only has a topic. To generate specific words, the **MD2** associating topics to words is adopted to develop ***K*** **DD2** (***K*** is the number of topics), and each **DD2** determines specific words of a certain topic (Blei et al., 2003; Newman, 2007). Eq. 1 describes the document generation process, which is also shown in Figure 5.

$$\prod_{j=1}^{M} P(\theta_j;\alpha) \prod_{i=1}^{K} (\phi_i;\beta) \prod_{n=1}^{N} P\left(Z_{j,n}|\theta_j\right) P(W_{j,n}|Z_{j,n}) \quad \text{Eq. 1}$$

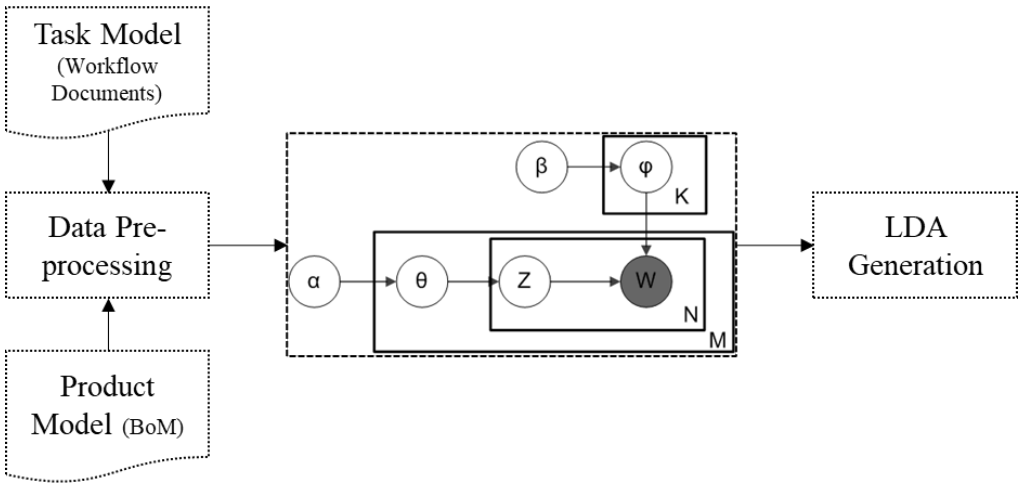


Figure 5. LDA document (TAs) generation process

This study borrows the above ideas, except that: 1) task assignments (TAs) (more details are introduced below) are generated to replace the role of documents; and 2) the task types and products are treated as topics and words, respectively. The **DD1** decides probabilities that a TA belongs to each task type, and the **DD2** decides the probabilities of building products used in a task given the task type. In other words, in this study, the ***M***, ***K***, and ***N*** in Eq. 1 represent the number of TAs, task types, and products in a TA, respectively. Figure 5 shows the flow of the LDA model. The training data come from two sources: 1) the MC production workflow, where each task is extracted as a unique task type, and 2) the BoM, where each item is taken as a product. Then, the products are manually associated with task types, generating task assignments (TAs) as inputs to train the LDA model. Each TA includes one task type and a set of building products, and Table 1 lists several examples.

Training the LDA model requires estimating the parameters α and β so that the model can maximize the probability of generating similar TAs in the training data. This can be realized by Gibbs sampling. The method works as follows: 1) randomly assigns a task type to a building product in the training data, 2) summarizes the number of products belonging to each task type in each TA and the number of products belonging to each task type in the entire dataset, 3) for each product, recalculates the **MD2** based on statistics obtained in the last step, 4) reassigns a new task type to each product using the newly obtained **MD2**, 5) repeats the process until parameters in the **MD1** and **MD2** converge (Gelfand, 2000). The maximum number of iterations is usually defined to save time and computation power. After training, each task type can be represented by key building products ranked by **MD2** distributions (see examples in Section 4). The aim is to map building products to tasks, which does not consider TA generation, only the **DD2** produced by training the LDA model is employed in model testing.

3.2.2. Data disambiguation

A challenge to implement the proposed model is that the names of tasks and products in the MC workflow and BoM are often freely written by engineers. This can cause ambiguity, e.g., different words are used to describe the same task. Thus, as suggested in Wu et al. (2021b), a heuristic method is proposed to 1) generate standard expressions (SEs) for products and tasks during training; and 2) match new products and tasks to the SEs during testing.

Given that the number of unique products for fabricating modules is not very large, the ambiguity of products' names often comes from their variations (e.g., studs and stud) and different orders of words when the name includes multiple words. Hence, pre-processing is applied (See example in Figure 6), which includes four steps: normalization (e.g., lowercasing), tokenization (i.e., separating a name into words), stop words removal (i.e., removing meaningless words), and stemming (i.e., converting a word to its basic form). The process is similar to text pre-processing, and more details can be found in Denny and Spirling (2016). The processed products during training form a large corpus covering SEs of common MC products. However, it is difficult to distinguish tasks in this way as their names contain many variations. For instance, a task can be carried out in different locations (e.g., installing studs at the wall or ceiling) or with different products/materials (e.g., installing cabinets or conduits). Therefore, based on practical experience, relevant literature, and the three ontologies, a lexicon is manually defined to cover tasks for fabricating MC modules. Each item in the lexicon is formed by ontology classes in the MPO, MTO, and PTO, which can unambiguously define a task type. Examples in the lexicon are shown in Table 2.

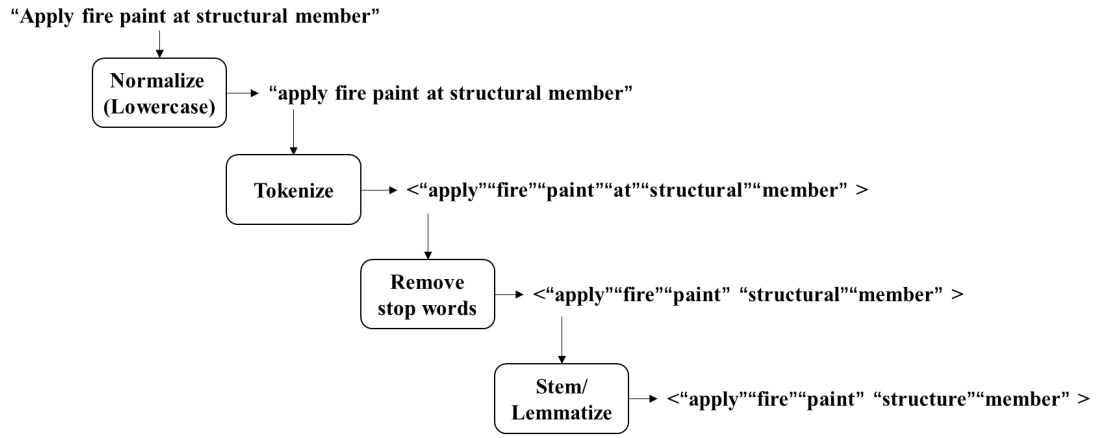


Figure 6 Example of four steps in data pre-processing

A similarity matching mechanism is carried out in the testing stage to find a standard expression for new products and tasks. The four-step mechanism is based on word

embeddings, where a new product: 1) undergoes the pre-processing steps introduced; 2) traverses the corpus or lexicon; 2) computes similarity value among the pre-processed new tasks/products, each SE in the corpus, and lexicon following Eq. 2, where $\boldsymbol{e^c}$ and $\boldsymbol{e^e}$ are word embeddings of a token of the SE and new products/tasks, respectively; 3) obtains an average similarity $\boldsymbol{s}$; 4) replaces the new products and tasks using the SE with the maximum $\boldsymbol{s}$. Word embeddings are employed as they are low-dimension real-valued vectors that can capture meanings of words despite different expressions (Church, 2017). Word embeddings should be learned by machine learning models (e.g., word2vec). However, training such models is very data demanding. Thus, it is common to utilize word embeddings trained by others. Table 1 lists several examples to show the effect of disambiguation.

$$\boldsymbol{s} = \frac{\sum_{i=1}^{n} e_i^c \times e_i^e}{\sqrt{\sum_{i=1}^{n} (e_i^c)^2} \times \sqrt{\sum_{i=1}^{n} (e_i^e)^2}} \quad (\boldsymbol{n}\text{=the dimension of the word embedding}) \quad \text{Eq. 2}$$

3.2.3. Data enriching based on spatial relations

Following the conventional approach to develop LDA models, a TA only consists of associated products. However, such simple data cannot suit situations in practical MC projects. One common problem is that when producing a room module, a product is required in multiple tasks thus should be included in multiple TAs. For instance, a gypsum board simultaneously belongs to the tasks 'board pre-treatment and punching' and 'installation of the board at wall surface'. Therefore, the conventional LDA model cannot distinguish such differences and can only make a random guess when mapping a new 'gypsum board' product.

To address this issue, spatial relations of module products are identified by referring to the BIM model and MTO to enrich TA data. The process has three steps: 1) as the BoM is derived from the BIM model, associated spatial instances (i.e., instances of MTO classes) of each product can be extracted from the BIM model (e.g., a gypsum board is related to a 'wall' instance in BIM); 2) a 'contains' spatial relation is set up between the spatial instance and product; and 3) the triple taking the form (i.e., 'spatial instance contains product') is added to the TA data. If no spatial relation is found, for instance, the product is used in pre-installation tasks (e.g., board punching), a triple 'non contains product' is added. As such, the same products can be distinguished by different spatial relations with a spatial instance. Figure 7 illustrates the process and difference between TA data before and after enriching.

Moreover, after investigating the workflow charts and schedules of MC projects in Hong Kong, it is found that 1) the unit for managing MC projects is individual rooms (e.g., house modules); 2) distinct spatial instances are considered as one system, and tasks performed on these instances are managed as one package (e.g., installing studs in all walls of a room is treated as a single package). Therefore, it is unlikely that the same triple (e.g., 'wall contains

studs') appears multiple times in the TA data and confuses the model.

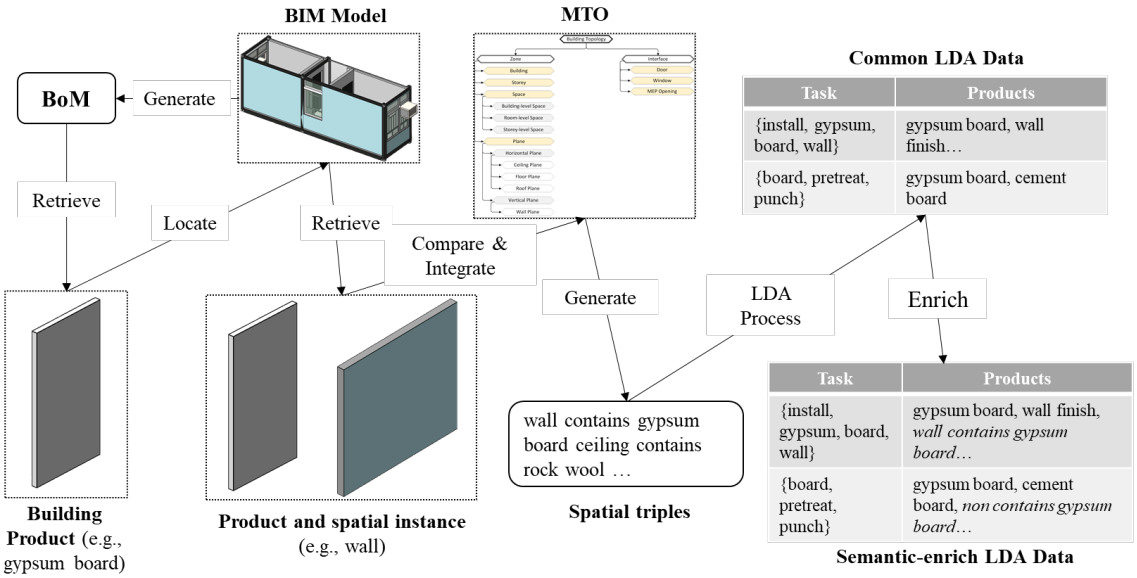


Figure 7 TA data enriching process

Table 1 Examples of TA data (before enriching)

Task (before disambiguation)	Product (before disambiguation)	Task (after disambiguation)	Product (after disambiguation)
Stud installation at wall	Metal skeletons, Shear Studs, RUnner, the panel of wall, TRACK, channels, wall Tiles, some noggins, braces	{stud instal wall}	metal skeleton, shear stud, runner, panel wall, track, channel, wall tile, noggin, brace
3D assembly	2D Panels, Corner Casting, Steel COLUMN, Steel Angles, Shear StuDS, CandleLoc System, ceiling beam, Purlins, FLOOR beam	{3d assembl}	2D panel, corner cast, steel column, steel angle, shear stud, candleloc system, ceiling beam, purlins, floor beam
Door ironmongery and windows installation	Ironmongery of DOOR aluminium window, FRP Timber door, glass PANEL, Hinges, hanDLes, latches	{door, ironmongeri, window, instal}	ironmongery door, aluminum window, frp timber door, glass panel, hinge, handle, latch

Table 2 Examples in the standard task lexicon

Task	Task pre-processed	Product lexicon	MTO	MPO	PTO
2D panel assembly	{2d assembl}	{Panel, Assembling}	N/A	Panel	Assembling
Stud installation at wall	{stud instal wall}	{Wall, Stud, Installing}	Wall	Stud	Installing
Door and window frame installation	{door window frame instal}	{Door, Window, Door Interface, Window Interface, Installing}	{Door Interface, Window Interface}	{Door, Window}	Installing

3.3. Task-work package mapping using weighted K-means clustering

3.3.1. Ontology-based task numerical presentation – Task2vec

The automated task-package mapping depends on clustering MC tasks, which requires transforming tasks to numerical vectors according to their attributes. Based on literature review and discussion with engineers in MC projects, three critical attributes are identified when packaging tasks: task relations, spatial relations, and resource demands (Li and Hall, 2019).

Task relationships concern sequential (preceding and succeeding), parallel, and coupled relations between two tasks. Figure 8 illustrates the three relationships. In particular, the coupled relation indicates that two tasks are interchangeably performed, such as '2D panel assembly' and 'butt and fillet welding'. When packaging tasks, planned schedules are often available. Thus, tasks for fabricating a module can be indexed from 1 to $\boldsymbol{T}$ based on their dependencies. Then, the dependency vector $v^d \in \mathbb{R}^3$ can be generated, where each entry is the index of another task holding the preceding, succeeding, and parallel relation with it (the value is -1 if no parallel task is found). It should be noted that the coupled relation is not explicitly covered in v^d, since it can be modeled if two tasks are indexed as predecessors and successors of one another simultaneously.

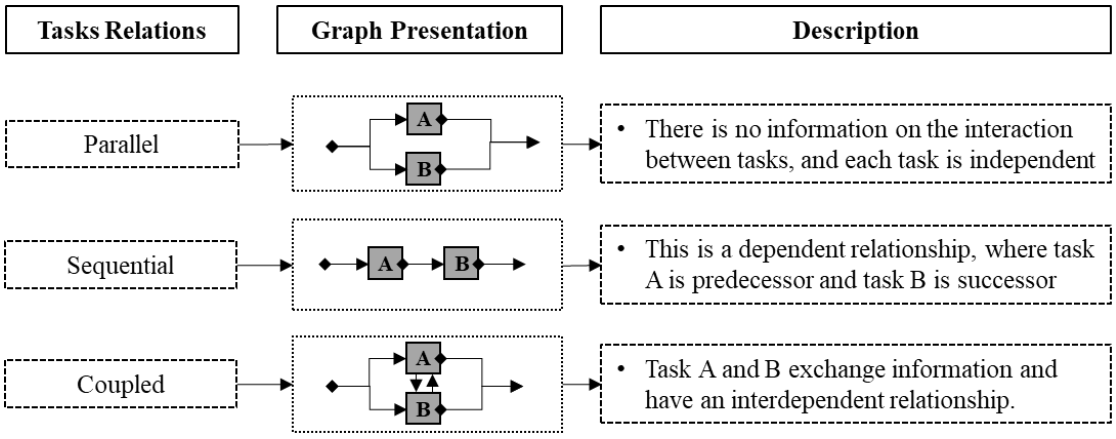


Figure 8 Tasks relations

The spatial attributes of a task are determined by the spatial instances and module products mapped to the task. Hence, a spatial vector v^s is generated using a one-hot encoding approach as follows: 1) for each task, v^s is initialized as a zero vector of shape $\in \mathbb{R}^C$, where each entry corresponds to one of the $\boldsymbol{C}$ classes in the MTO; 2) the bottom-level spatial instance of each product belong to the task is extracted by referring to spatial triples introduced previously (e.g., 'ceiling' is extracted from the triple 'ceiling contains rockwool'); 2) a spatial class list is constructed by searching for superclasses in the MTO, e.g., the list {Zone, Space, Room-Level Space, Horizontal Plane, Ceiling} can be formed given the

instance ‘ceiling’, while the list {Interface, MEP Opening} can be formed given ‘conduits’; 3) a complete list is generated by integrating information from lists of all products of the task; 4) v^s is re-evaluated, where the entry value is set as one if the corresponding ontological class exists in the complete list. Locations (i.e., local coordinates) of all products assigned to the same task should be the same in an MC module (e.g., studs and tiles consumed in one task ‘wall stud installation’ are located in the same place). Thus, generating v^s needs to only consider one product, but the above process considers all products to avoid missing information (Salama et al., 2017). On the other hand, many MC tasks require distinct material resources, e.g., painting materials for painting and coating tasks and welding materials for assembling tasks. As such, the resource vector $v^r \in \mathbb{R}^D$ can be formed following a similar procedure to generating spatial vectors, where $\boldsymbol{D}$ is the number of classes in the MPO. The three vectors (i.e., v^d, v^s, v^r) are concatenated into a single long vector $v^t \in \mathbb{R}^{3+C+D}$ to convert an MC task into a numerical vector. Figure 9 presents the pseudo-code of the process.

Algorithm 1 Task2vec process

Input: ontologies $\boldsymbol{MPO}$, $\boldsymbol{MTO}$, task relationships $\boldsymbol{TR}$, task assignments $\boldsymbol{TA}$
Onput: task vector v^k

for t=1,2$\cdots$ $\boldsymbol{T}$ **do**
  initialise $v^d \in \mathcal{R}^3$
  preceding task index $p^t \xleftarrow{searching} \boldsymbol{TR}$
  succeeding task index $s^t \xleftarrow{searching} \boldsymbol{TR}$
  parallel task index $c^t \xleftarrow{searching} \boldsymbol{TR}$
  if c^t not exists **then**
    c^t=-1
  end if
  encoding the 3 entries in v^d using p^t, s^t, c^t, respectively
end for

for t=1,2$\cdots$ $\boldsymbol{T}$ **do**
  product set $\boldsymbol{P} \xleftarrow{extract\ building\ products} \boldsymbol{TA}$
  $\boldsymbol{C}, \boldsymbol{D} \xleftarrow{get\ the\ number\ of\ classes} \boldsymbol{MTO}, \boldsymbol{MPO}$
  initialise $v^s \in \mathcal{R}^C; v^r \in \mathcal{R}^D$
  for i $\in \boldsymbol{P}$ **do**
    $\boldsymbol{s} \xleftarrow{extract\ spatial\ product} \boldsymbol{TA}$
    material class $list_m^i \xleftarrow{searching\ \boldsymbol{s}} \boldsymbol{MPO}$
    spatial class $list_s^i \xleftarrow{searching\ \boldsymbol{s}} \boldsymbol{MTO}$
  end for
  complete material class $list_m^t = \bigcup list_m^i\ i = 1, 2\cdots$
  complete spatial class $list_s^t = \bigcup list_s^i\ i = 1, 2\cdots$
  $v^r = one - hot\ encoding(list_m^t)$
  $v^s = one - hot\ encoding(list_s^t)$
  $v^t = concat(v^d, v^s, v^r)$
end for

Figure 9 The pseudo-code of task2vec process

3.3.2. K-means clustering with heuristic weighting

K-means clustering is a simple but effective unsupervised algorithm to automatically partition ***m*** data samples into ***L*** clusters. It works as an iterative refinement process: 1) randomly select ***L*** samples as initial centroids, 2) compute the distance (e.g., Euclidean distance) between the cluster centroids and all other samples; 3) assign each sample to the cluster with the nearest distance; 3) recalculate cluster centroids using all samples it contains then reassign all samples. The process converges when the assignments no longer change or after a certain number of iterations (Tsai and Chiu, 2008). Steps 2-3 are shown in Eq. 3, which determines the cluster of the i^{th} task according to centroids u_j $\{j = 1,2 \dots \boldsymbol{L}\}$.

$$c^{(i)} := \underset{j}{\operatorname{argmin}} \left\| x^i - u_j \right\| \quad \text{Eq. 4}$$

However, K-means clustering can cause many errors if it is directly applied to task vectors. For instance, the two tasks 'door frame installation' and 'door ironmongery installation' will be grouped in one cluster (i.e., one package) despite that in practice, they are separated by a dozen tasks and managed in two packages. Most parts of the vectors (i.e., v^s, v^r) of the two tasks are similar owing to similar class lists. Therefore, the differences between v^d cannot impose its impact. On the other hand, it is critical to ensure enough classes are covered in v^s and v^r, otherwise closely related tasks can be grouped in different packages. To address this problem, before clustering, vectors v^s, v^r are normalized (i.e., each sample subtracts the mean and is divided by the standard deviation of all samples), while v^d remains unchanged. In this way, the K-means algorithm can pay more attention (in other words, assigns more weights) to the dependency attributes of data. In addition, parallel tasks should not be packaged, to consider this rule in the clustering process, a penalty λ is added when computing the distance between tasks, where a large value (e.g., 10) is assigned to λ if a parallel task of the current task $x^{(i)}$ is found to exist in a cluster.

Another issue is that the one-hot encoding can result in similar vectors where most entries are 0, making clustering difficult. Therefore, a heuristic weighting method is proposed. It is reasonable to assign more weights to more abstract classes while fewer weights to more specific classes. For instance, installing windows and doors should be packaged together because they are instances of installing 'Furniture', although they have different bottom-level classes (i.e., 'Window' and 'Door') in the class lists. As such, descending integer weights are assigned based on class levels in the MPO and MTO, e.g., weights {4, 3, 2, 1} are assigned to the level 1 to 4 classes in the MPO, which results in a weight vector ***w***. Task vectors are multiplied with ***w*** through Hadamard production before being fed to K-means, making differences among vectors more distinct. The pseudo-code of the weighted K-means

clustering is shown in Figure 10.

Algorithm 2 weighted K-means

Input: task vectors $v_1, v_2 \cdots v_{\boldsymbol{V}}$, ontology class weight vector $\boldsymbol{w}$, the number of clusters $\boldsymbol{L}$, the maximum iteration number $\boldsymbol{O}$
Onput: cluster labels for each task $l_1, l_2 \cdots l_{\boldsymbol{T}}$

for t=1,2$\cdots \boldsymbol{T}$ **do**
  $v^d, [v^s, v^r] \xleftarrow{extract} v^t$
  $[v^s, v^r] = normalise([v^s, v^r] \circ \boldsymbol{w})$
  $v^t = concat(v^d, [v^s, v^r])$
end for

initialise random $\boldsymbol{L}$ cluster centroids $u_1, u_2, \cdots u_{(\boldsymbol{L})}$
initialise cluster assignment $c^{(i)} := \arg\min_l \parallel x^{(i)} - u_l \parallel$
for o=1,2$\cdots \boldsymbol{O}$ **do**
  $c^{(i)} := \arg\min_l \parallel x^{(i)} - u_l \parallel + \lambda$
  $\lambda = \begin{cases} 10 & \text{parallel}(x^{(j)}) = \text{parallel}(x^{(i)})\ x^{(j)} \in c^{(i)} \\ 0 & \text{otherwise} \end{cases}$

  $u_l := \frac{\sum_{i=1}^{m} 1\{c^{(i)}=j\} x^{(i)}}{\sum_{i=1}^{m} 1\{c^{(i)}=j\}}$
end for

Figure 10 The pseudo-code of weighted K-means clustering

Finally, the number of clusters $\boldsymbol{L}$ (i.e., the number of packages) must be determined, and this study uses one of the most popular methods, i.e., the gap statistic (GP) method, to find the optimal $\boldsymbol{L}$. The K-means model can be evaluated by the distortion value, i.e., the sum squared error (SSE) of the distance between each data sample and its assigned centroid (see Eq. 5). The GP method could find $\boldsymbol{L}$ automatically, which works as follows: 1) generates random data samples using uniform distribution; 2) applies K-means to the generated samples and compute SSE; 3) repeats above two steps multiple times (often through Monte Carlo simulation) and obtain the SSE; 5) computes the GP value using Eq. 6; 4) determines the optimal $\boldsymbol{L}$ as the one obtaining the largest GP value. More details of the GP method can be found in Yan and Ye (2007).

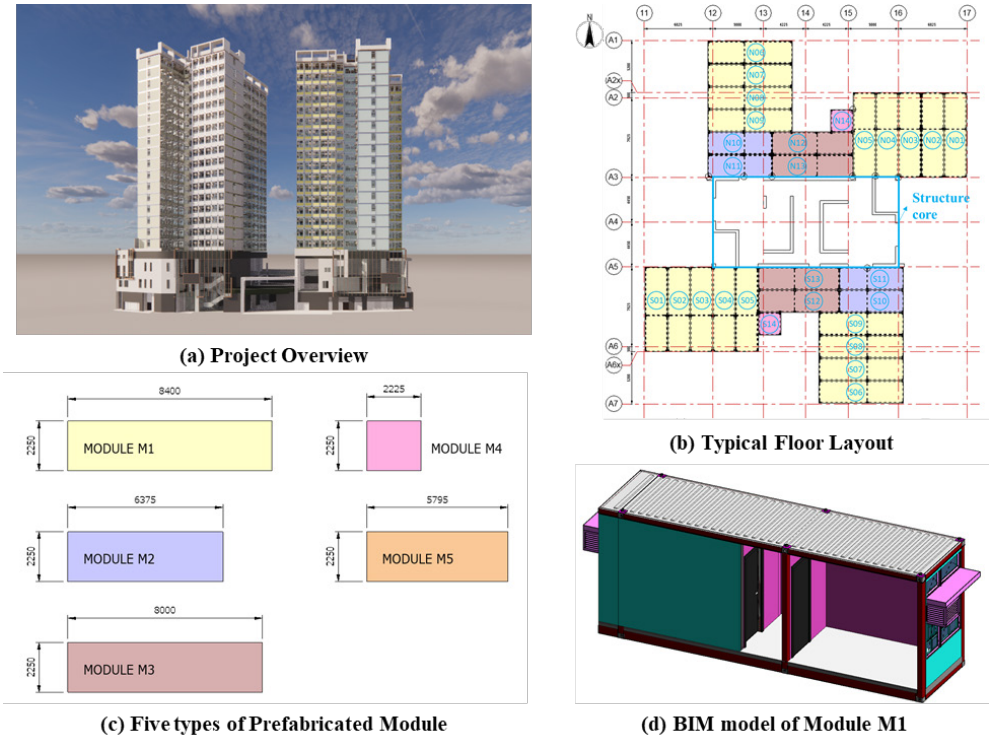
$$SSE = \sum_i^m \left\| x^i - u_i \right\| \qquad \text{Eq. 5}$$

$$Gap(K) = E(\log SSE_k) - \log SSE_k \qquad \text{Eq. 6}$$

4. Experiments

In this section, experiments are conducted to evaluate the performance of the DOM model for work package generation in a real MC project in Hong Kong. This project is a student residence with two 17-floor towers comprising 1224 prefabricated hostel rooms and other

supporting facilities (e.g., prefabricated toilet, kitchen). The graphical details of the project are shown in Figure 11. All the prefabricated modules are produced at an off-shore factory in Mainland China, and this project is currently under the mock-up stage. As this factory will simultaneously produce around 400 modules with a peak of 600 workers at the mass production stage, automatically generating semantic-enrich work packages under the complex production environments for each worker or team are urgently needed. DOM proposed in this study has the potentials to facilitate collaborative and dynamic planning in the mass production stage of MC.



(a) Project Overview

(b) Typical Floor Layout

(c) Five types of Prefabricated Module

(d) BIM model of Module M1

Figure 11 Overview of the MC project for this experiment

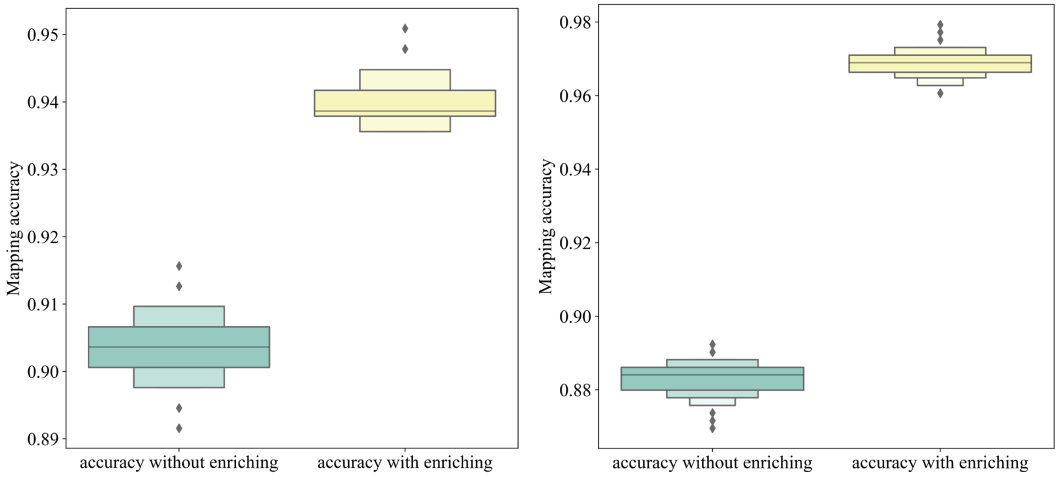
4.1.Data preparation

As the project went on, more than ten versions of workflow/schedule and BIM model (i.e., can generate BoM) had been generated and updated for producing module M1. The number of products in BoM and tasks in the workflow increased as the schedule went on. The initial version (T-10) and the one for the latest mock-up stage (T-23) are selected for this experiment, which includes 10 tasks and 23 tasks, respectively. The changes in these two versions of workflow and BoM can lead to changes in work package content. Thus it can not only help test the dynamic performance of the DOM model in work packaging but also prove that the proposed method can be repeatable in varied datasets.

4.2.Results Analysis

4.2.1. Product-task mapping results

Figure 12 compares the effect of disambiguation and data enriching for both T-10 and T-23. More errors occur if without enriching. After running the model 100 times, the average accuracy is 0.904 and 0.938 for T-10, 0.883 and 0.968 for T-23, respectively. The results show that the spatial relation enriching could significantly improve the mapping accuracy between products and tasks.



(a) T-10 (b) T-23

Figure 12 Accuracy of product-task mapping

The errors mainly come from the lack of information for the LDA model to classify products. Figure 13 shows a few distinct examples, where the counts are obtained by running the naive LDA model 100 times while counting the task assignments of the products in the T-23 situation. Some products can only be used in a particular task (e.g., electrodes for the welding task), which can be handled by the navie LDA model. However, several products (e.g., ‘promatect h-board’ and ‘ceiling panel’) can belong to two or more tasks. The same ambiguation happened when mapping products ‘shear stud’ and ‘gypsum board’ in the T-10 situation. The navie LDA model cannot distinguish such product-task relations, thus make wrong assignments.

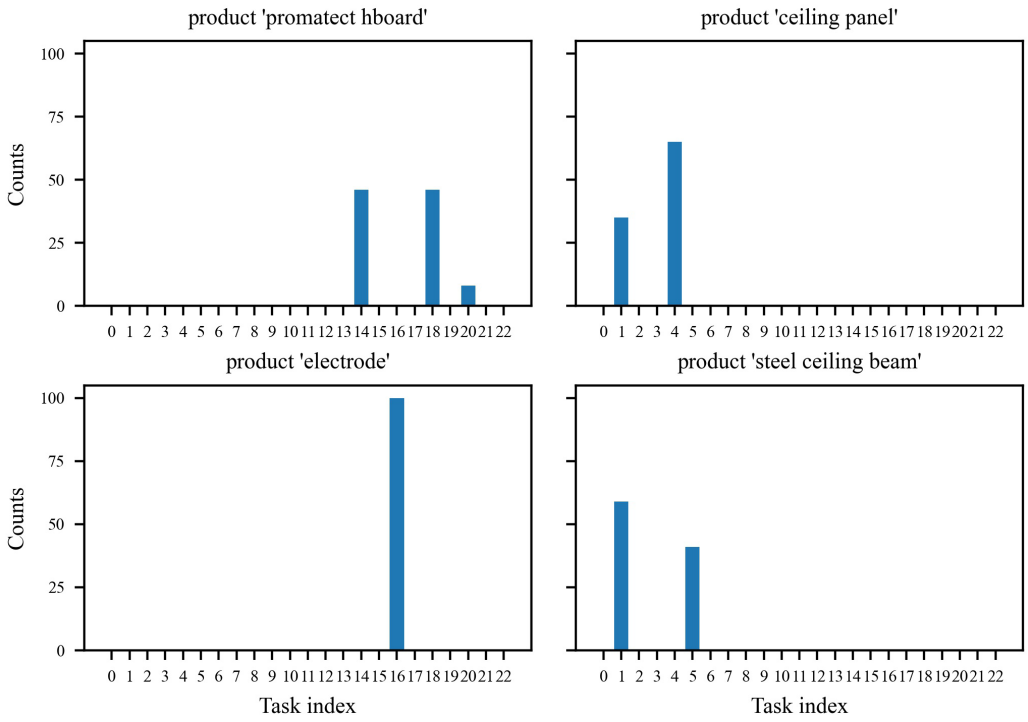


Figure 13 Examples of errors that occurred without data enriching

The confusion matrices in Figure 14 present the differences between products with data enriching and without data enriching. For instance, for T-23, the product ‘steel column beam’ can belong to either ‘3D assembly’ (task 5) or ‘installing ceiling studs’ (task 1). The matrix in Fig.14 (c) shows that seven ‘steel column beam’ products are wrongly assigned to task ID-1. The situation is much better in the second matrix, and all ‘steel column beam’ products can be correctly assigned to task 1.

However, errors still occur for a few products that cannot be distinguished even when spatial relations are utilized. For instance, for T-23, the product ‘ceiling panel’ should be used in task 9, ‘installing studs at the ceiling’, as stated in the workflow. However, the LDA model wrongly assigns it to task 15‘layer and coat painting ceiling’ because the product has the same spatial relations in both tasks. Nevertheless, the accuracy for T-23 is still increased by more than 7%.

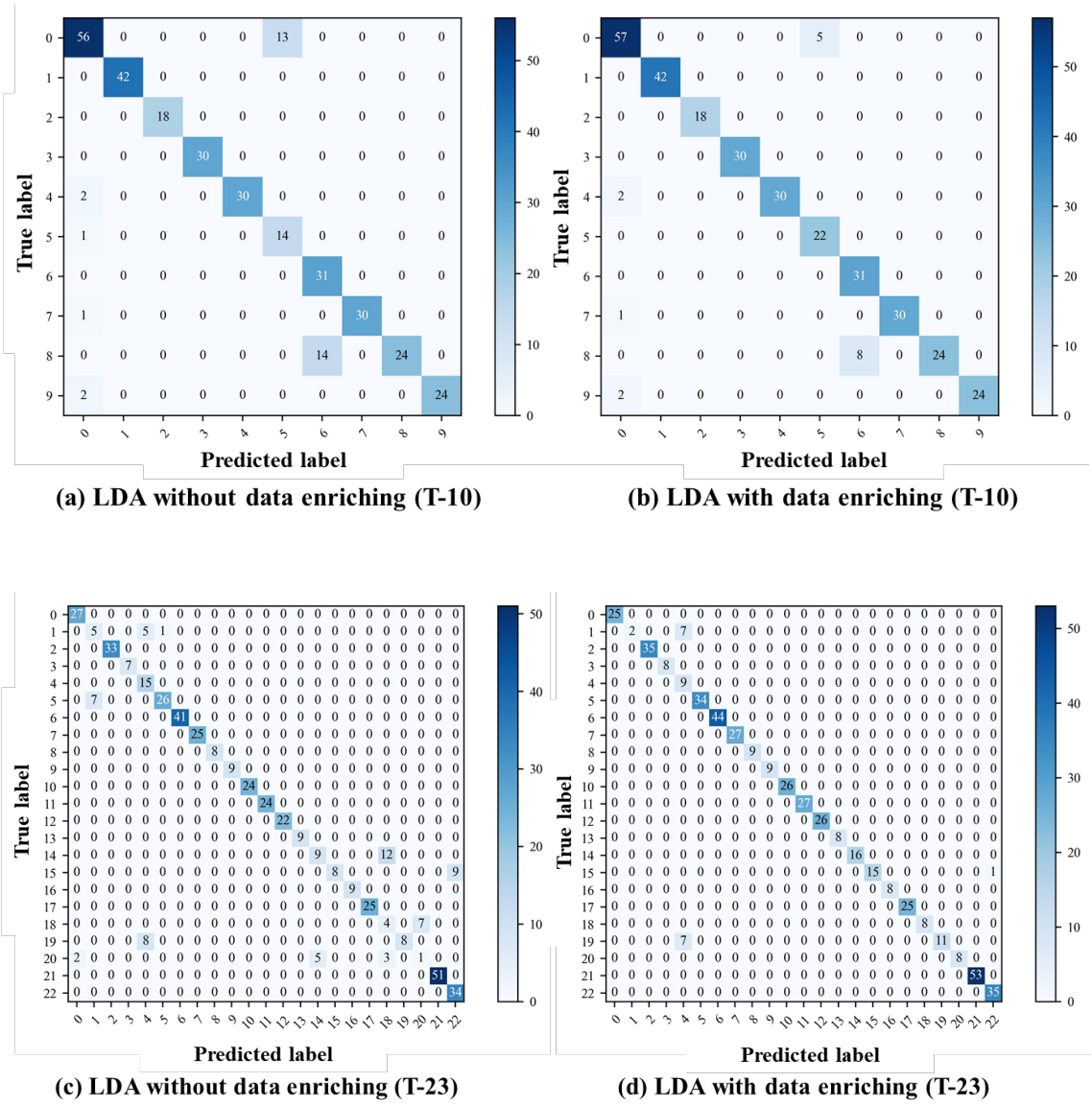


Figure 14 Confusion matrix of product-task mapping (both x and y-axis are task ID, see Table 3 & 4 for specific task names, the tasks are not arranged in sequence in the product-task mapping stage)

4.2.2. Task-package mapping results

In this section, the weighted K-means model is demonstrated when packaging tasks for producing module M1. Task vectors are multiplied with weight vector $\boldsymbol{w}$ through Hadamard production before being fed to K-means, making differences among vectors more distinct. The optimal number of packages is determined using the GP method, automatically selecting L, thus minimizing subjective judgment. As shown in Figure 15, the method suggests six and ten work packages as the good choice for T-10 and T-23, respectively.

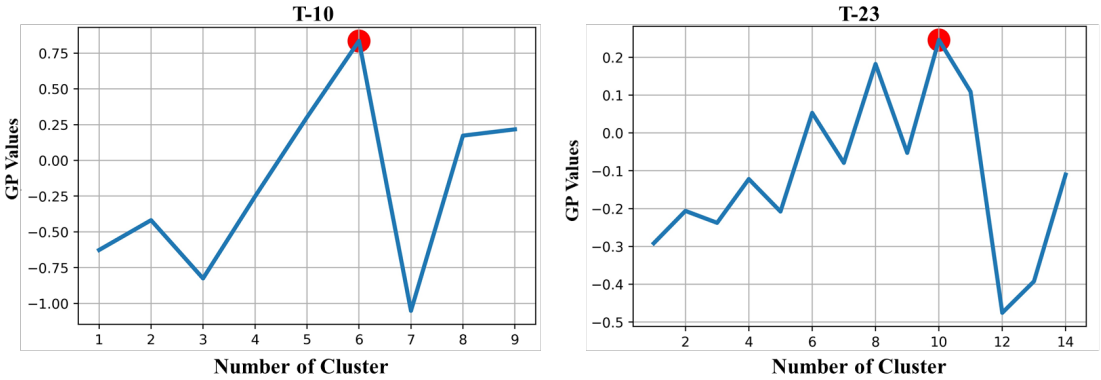


Figure 15 Determining the optimal number of clusters (work packages)

Tables 3 & 4 compare the manual assignment results (based on the project manager's experience) and three K-means mapping results for T-10 and T-23, respectively. MA and KM mean assigning MC tasks manually and through k-means clustering, respectively. KM-1 does not implement heuristic weighting as introduced in Section 3.2.3. In Table 3, the proposed clustering method perfectly packages the tasks, the same as manual packaging results. However, KM-1 wrongly packages tasks 7-9, where tasks 7 and 9 are separated by task 8, which is impractical given their close dependencies. It is because they have similar spatial and resource attributes, which outweigh the impact of task relationships when heuristic weighting is not applied. In Table 4, KM-1 for T-23 does not separate tasks 2-4 as attribute vectors (one-hot encoding without heuristic weighting and normalization), including many 0 and a few 1, which are not distinct enough to be separated. Moreover, similar to the issue in Table 3, the two tasks (i.e., tasks 8 and 23) are packaged together in KM-1, despite being separated by many tasks. The packaging results for tasks 12-14 have a similar problem.

In contrast, when handling the 23 tasks, the proposed K-means model can address both issues and produce the most practical packaging results, which are also very similar to the manual packaging results. The only difference is that fabricating the wall and ceiling is separated into two packages instead of one. It is because the manual packaging suggests nine packages, while the K-means model suggests ten. It is also reasonable as tasks for wall and ceiling can be separated by the spatial relations, and they can also be executed concurrently and combined as the wall work package in practice.

Table 3 Comparison of task-package mapping results

ID	Task	MA	KM-1	Proposed KM
0	Board pretreatment and punching	1	1	1
1	2D panel assembly	2	2	2
2	Butt weld and fillet weld	2	2	2

3	3D assembly	3	3	3
4	Apply fire paint at strucutral member	3	3	3
5	Rebar Fixing	4	4	4
6	Pouring and curing Concrete	4	4	4
7	Door and window frame installation	5	5	5
8	Stud installation at the ceiling	6	6	6
9	Stud installation at wall	6	5	6

Table 4 Comparison of task-package mapping results

ID	Task	MA	KM-1	Proposed KM
0	Board pretreatment and punching	1	1	1
1	2D panel assembly	2	2	2
2	Butt weld and fillet weld	2	2	2
3	3D assembly	3	2	3
4	Apply fire paint at strucutral member	3	2	3
5	Rebar Fixing	4	3	4
6	Pouring and curing Concrete	4	3	4
7	Door and window frame installation	5	4	5
8	Installation of MEP at the ceiling	6	6	6
9	Stud installation at the ceiling	6	5	6
10	Rockwool in-fill and fire board installation at the ceiling	6	5	6
11	Stud installation at wall	6	5	7
12	Rockwool in-fill and fire board installation at wall	6	5	7
13	Installation of MEP at the wall surface	6	6	7
14	Installation of gypsum board at the wall surface	6	5	7
15	Layer and coat painting ceiling	7	7	8
16	Layer and coat painting wall	7	7	8
17	Installation of pipes	8	8	9
18	Electrical cable wiring	8	8	9
19	Window type AC installation	9	9	10
20	Installation of lighting system	9	9	10
21	Cabinets installation	9	10	10
22	Door and window ironmongery	9	4	10

505

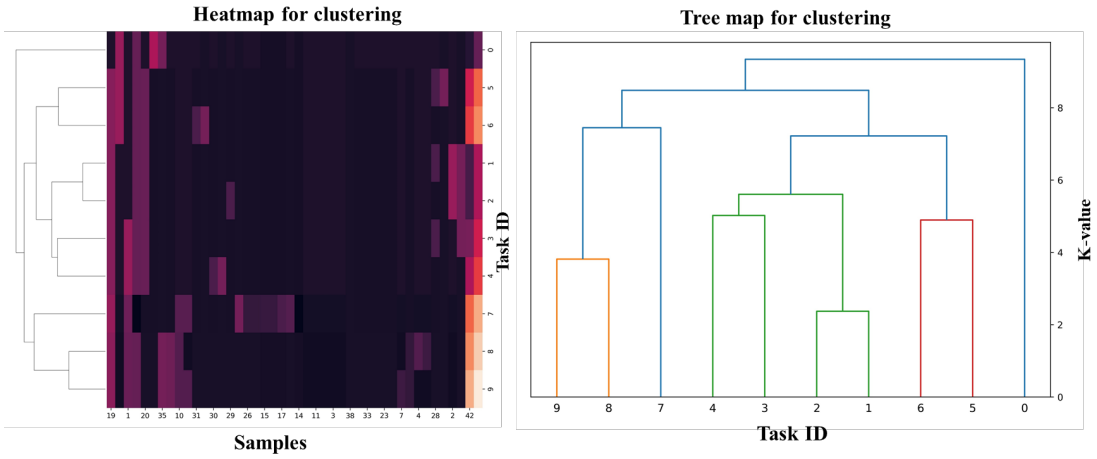
Finally, the proposed K-means model was applied to all 368 tasks in this case project. The

model was evaluated using two important metrics (See Table 3 & 4), i.e., the Silhouette Coefficient and Calinski-Harabaz index. The two metrics evaluate the SEE and data co-variance within each cluster (work package) and between clusters (work packages), and both favor the large values. It can be argued that the proposed weighting method can significantly improve the model performance. In addition, the improvement is more evident with more tasks (i.e., when the situation is more complex).

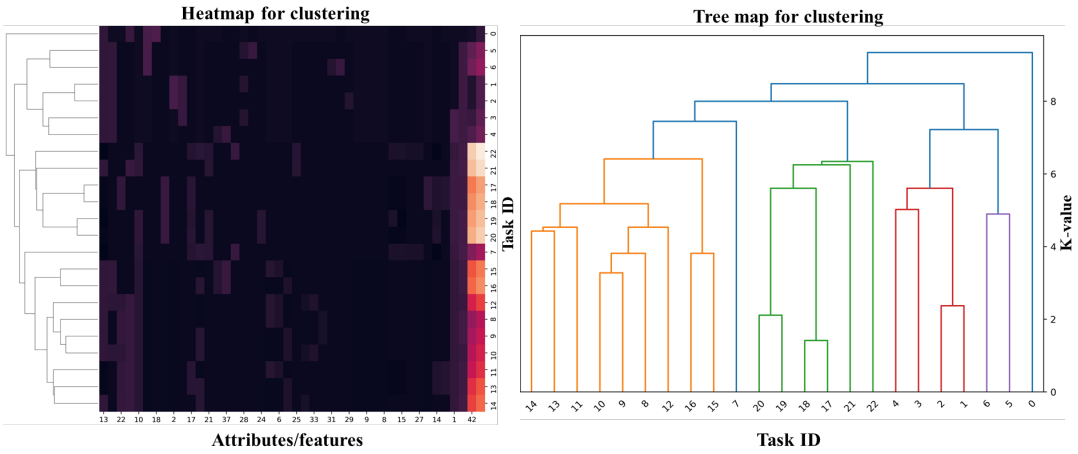
Table 4 Comparison between K-means clustering metrics

Task	KM-1	KM (proposed)
Silhouette Corfficient (T-10)	0.582	0.855
Calinski-Harabaz index (T-10)	307.057	548.161
Silhouette Corfficient (T-23)	0.440	0.569
Calinski-Harabaz index (T-23)	97.196	487.660

Figure 16 demonstrates the clusters (i.e., packaging results). Task ID in Fig.16 could be well matched with Tables 3 & 4. Each rectangular area with the same color suggests rows correlated with the corresponding columns. Tasks in both T-10 and T-23 can be separated into distinct clusters (i.e., working packages). The right two columns for both T-10 and T-23 have lighter colors. Because vectors v^s, v^r are normalized, while v^d remains unchanged in these two samples. In this way, the K-means algorithm can assign more weights to the dependency attributes of data.



(a) T-10



(b) T-23

Figure 16 Overview of K-means clustering results

5. Discussion

The proposed DOM model is a brand new approach to enrich the semantics of tasks and generate the work packages more accurately and automatically. Compared with the previous studies, three aspects of the DOM model's novelty are summarized as follows.

- Firstly, three ontology models, namely, MPO, MTO, and PTO, are established as the knowledge base in MC, which facilitates extracting information in product, topology, and tasks. These ontology models represent not only the knowledge of product breakdown structure, products' spatial topology, and work breakdown structure but also form relationships between these knowledge domains. Previous works, such as An et al. (2019), have built the knowledge domains in MC's resource, operation, and product. However, they only modeled knowledge for wood frames rather than an entire prefabricated module and considered less knowledge in the product topology.

- Secondly, the improved LDA model enriches products with spatial topology relations to improve the mapping accuracy between products and tasks. LDA is a kind of unsupervised machine learning technique that does not consider the relations of different retrieved words, and it may confuse the model when the same product name under different statuses should be mapped to more than two tasks (See Fig.7). MTO can enrich the product data to improve the efficiency and accuracy of product-task mapping, and these enriched product data can play an important role as the resource attributes in the work packaging process.

- Thirdly, the approach for forming a semantic-rich task by transforming a task's attributes of dependency, topology, and resources into a vector is first developed. A

heuristic weighting method is then designed to improve the clustering accuracy of K-means, which help finally packaging the semantic-rich tasks and get the practical work packages. The previous studies used the design structure matrix to generate final work packages. However, it only considers the task dependency and works manually (Lee et al., 2017).

Despite these contributions, our study still has several limitations.

- Firstly, As the ontology modeling process is extremely time-consuming, this study adopted the "transferred modeling" strategy by customizing and modifying existing ontology models (e.g., IfcBuildingElement, W3C building topology). Although it would be easy to expand the proposed ontology models to other MC projects, to make DOM more practical, it still requires sufficient knowledge to involve all relevant knowledge domains regarding module production and the more well-defined relationships among the three ontology models.

- Secondly, the modified LDA model in this study still requires a proper level of details (LoD) for products and tasks to maximize mapping accuracy. Compared with the naïve LDA model embracing Dirichlet distributions' inability to capture correlations, the proposed model can to-some-extent distinguish module products accurately by leveraging spatial relationships. However, ambiguity issues still arise if the LoD is too high. For instance, if the task 'stud installation' is divided into first- and second-layer stud installation, the LDA model cannot correctly assign the product 'stud'. The underlying reason could be twofold. (1) the length of BoM is short, leading to the scattering of document-level word co-occurrences. It could be further improved by extending the BoM document with more semantics (e.g., spatial relationships, attributes). (2) Some task compositions are overlapping, meaning that the same product can be in multiple tasks. Therefore, the generated tasks are not independent and orthogonal. It may consider the hierarchical LDA by combing tasks in a hierarchy using a Nested Chinese Restaurant Process (NCRP).

- Thirdly, the work packaging process in this study only considers some critical attributes of products and tasks. However, this study did not determine the optimal work package size in clustering tasks into packages by linking with project performance (e.g., schedule, cost). Many factors influence work package sizes, such as workload, schedule estimation, economies of scale, cost of monitoring and control.

6. Conclusion

Modular construction (MC) has the great potential to put the construction industry forward to Construction 4.0 since it follows industrialized principles that could provide a collaborative

working environment among workers, robotics, and machines. However, current practice indicates that cross knowledge domain tasks in the MC factory still require dynamic and collaborative planning, particularly in the mass production stage. Thus, this study proposed a dynamic ontology-based mapping (DOM) model to help generate work packages dynamically and automatically. Firstly, three ontology models, namely, MPO, MTO, PTO, are established as the knowledge base to facilitate extracting the information of products, spatial relations, and tasks. Then, the customized LDA model is developed to accurately map module products (enriched by the spatial topology) to relevant tasks, where these products can serve as the resource attributes for tasks. Finally, semantic-enrich tasks with dependency, topology, and resources could be converted into vectors, where the weighted K-means clustering method is developed to combine these tasks into optimal work packages. The experiment is conducted on two versions of datasets (i.e., workflow with 23 tasks and 33 tasks), the results show that the DOM model can help dynamically generate the optimal work packages when schedule and BoM changed.

Modular construction companies can benefit from our work in several ways. First, they can better understand the mechanism by which various knowledge related to modular products, tasks, and work package generation. Second, work packages can be generated efficiently in the mass production stage to instruct the workers, machines, or robotics' tasks execution in a complex MC production environment. Third, they can use this proposed solution to generate work packages that significantly reduce their planning costs.

Several topics remain open for future research. First, more MC tasks related ontology models (e.g., production tools and machines, durations, costs) can be further established to enrich the knowledge base for more accurate work packaging; Second, the factors (e.g., level of details in products) that affect the accuracy of the LDA model for product-task mapping should be further investigated to improve LDA model with more semantic features, or other supervised learning model can be developed for product-task mapping; Third, we note that our work focuses on work package formation, rather on the work package sizing. Developing a more complicated model that simultaneously generates work packages and optimizes the work package sizes may be possible. In conclusion, we hope our work will encourage further research on this topic, potentially enhancing the collaborative planning and performance of MC projects.

Acknowledgment

This research was financially supported by the Hong Kong Innovation and Technology Fund (ITF) (No. ITP/029/20LP) and the Fellowship of China Postdoctoral Science Foundation (No. 2021M692169).

References

Al-Anazi, S., AlMahmoud, H., & Al-Turaiki, I. (2016). Finding similar documents using different clustering techniques. *Procedia Computer Science*, *82*, pp.28-34.

Doi: 10.1016/j.procs.2016.04.005

An, S., Martinez, P., Ahmad, R., & Al-Hussein, M. (2019). Ontology-based knowledge modeling for frame assemblies manufacturing. In *ISARC. Proceedings of the International Symposium on Automation and Robotics in Construction* (Vol. 36, pp. 709-715). IAARC Publications. Doi: 10.22260/ISARC2019/0095

Arashpour, M., Kamat, V., Bai, Y., Wakefield, R., & Abbasi, B. (2018). Optimization modeling of multi-skilled resources in prefabrication: Theorizing cost analysis of process integration in off-site construction. *Automation in Construction*, *95*, pp.1-9. Doi: 10.1016/j.autcon.2018.07.027

Aversano, L., Grasso, C., & Tortorella, M. (2016). Managing the alignment between business processes and software systems. *Information and Software Technology*, *72*, pp.171-188. Doi: 10.1016/j.infsof.2015.12.009

Baker, H., Hallowell, M. R., & Tixier, A. J. P. (2020). Automatically learning construction injury precursors from the text. *Automation in Construction*, *118*, 103145. Doi: 10.1016/j.autcon.2020.103145

Baskara, A. R., Sarno, R., & Solichah, A. (2016, October). Discovering traceability between business process and software component using Latent Dirichlet Allocation. In *2016 International Conference on Informatics and Computing (ICIC)* (pp. 251-256). IEEE. Doi: 10.1109/IAC.2016.7905724

Blei, D. M., Ng, A. Y., & Jordan, M. I. (2003). Latent Dirichlet Allocation. *The Journal of Machine Learning Research*, *3*, 993-1022. Doi: 10.5555/944919.944937

Church, K. W. (2017). Word2Vec. *Natural Language Engineering*, *23*(1), pp.155-162. Doi: 10.1017/S1351324916000334

Denny, M. J., & Spirling, A. (2018). Text preprocessing for unsupervised learning: Why it matters, when it misleads, and what to do about it. *Political Analysis*, *26*(2), pp.168-189. Doi: 10.1017/pan.2017.44

DOD and NASA. (1962). DOD and NASA Guide: PERT COST Systems Design. Retrieved from https://babel.hathitrust.org/cgi/pt?id=mdp.39015006057866&view=1up&seq=1 on

Ekanayake, E. M. A. C., Shen, G. Q., & Kumaraswamy, M. M. (2020). Identifying supply chain capabilities of construction firms in industrialized construction. *Production Planning & Control*, pp.1-19. Doi: 10.1080/09537287.2020.1732494

Eriksson, P. E., Olander, S., Szentes, H., & Widén, K. (2014). Managing short-term efficiency and long-term development through industrialized construction. *Construction Management and Economics*, *32*(1-2), pp.97-108. Doi: 10.1080/01446193.2013.814920

Gann, D. M. (1996). Construction as a manufacturing process? Similarities and differences between industrialized housing and car production in Japan. *Construction Management and Economics*, 14(5), 437-450. Doi: 10.1080/014461996373304

Gelfand, A. E. (2000). Gibbs sampling. *Journal of the American Statistical Association*, *95*(452), pp.1300-1304. Doi: 10.1080/01621459.2000.10474335

Goh, M., & Goh, Y. M. (2019). Lean production theory-based simulation of modular construction processes. *Automation in Construction*, *101*, pp.227-244. Doi: 10.1016/j.autcon.2018.12.017

Golpayegani, S. A. H., & Emamizadeh, B. (2007). Designing work breakdown structures using modular neural networks. *Decision Support Systems*, *44*(1), pp.202-222. Doi: 10.1016/j.dss.2007.03.013

Hammad, A. W., Akbarnezhad, A., Wu, P., Wang, X., & Haddad, A. (2019). Building information modeling-based framework to contrast conventional and modular construction methods through selected sustainability factors. *Journal of Cleaner Production*, *228*, pp.1264-1281. Doi: 10.1016/j.jclepro.2019.04.150

Han, K. K., Cline, D., & Golparvar-Fard, M. (2015). Formalized knowledge of construction sequencing for visual monitoring of work-in-progress via incomplete point clouds and low-LoD 4D BIMs. *Advanced Engineering Informatics*, *29*(4), pp.889-901. Doi: 10.1016/j.aei.2015.10.006

Hoffman, M., Bach, F., & Blei, D. (2010). Online learning for latent Dirichlet allocation. *Advances in Neural Information Processing Systems*, *23*, pp.856-864. Doi: 10.5555/2997189.2997285

Ibrahim, Y. M., Lukins, T. C., Zhang, X., Trucco, E., & Kaka, A. P. (2009). Towards automated progress assessment of work package components in construction projects using computer vision. *Advanced Engineering Informatics*, 23(1), pp.93-103. Doi:

10.1016/j.aei.2008.07.002

Isaac, S., Curreli, M., & Stoliar, Y. (2017). Work packaging with BIM. *Automation in Construction*, *83*, pp.121-133. Doi: 10.1016/j.autcon.2017.08.030

International Organization for Standardization (2021). *Construction materials and building* (ISO Standard No. 91). Retrieved from https://www.iso.org/ics/91/x/ on 14th July 2021

Jallan, Y., Brogan, E., Ashuri, B., & Clevenger, C. M. (2019). Application of natural language processing and text mining to identify patterns in construction-defect litigation cases. *Journal of Legal Affairs and Dispute Resolution in Engineering and Construction*, *11*(4), 04519024. Doi: 10.1061/(ASCE)LA.1943-4170.0000308

Jung, Y., & Woo, S. (2004). Flexible work breakdown structure for integrated cost and schedule control. *Journal of Construction Engineering and Management*, *130*(5), pp.616-625. Doi: 10.1061/(ASCE)0733-9364(2004)130:5(616)

Lawson, M., Ogden, R., & Goodier, C. (2014). *Design in modular construction*. CRC Press. ISBN 9780367865351

Lee, J., Deng, W. Y., Lee, W. T., Lee, S. J., Hsu, K. H., & Ma, S. P. (2010). Integrating process and work breakdown structure with design structure matrix. *Simulation*, *7*, 8. Doi: 10.20965/jaciii.2010.p0512

Lee, J., Park, M., Lee, H. S., Kim, T., Kim, S., & Hyun, H. (2017). Workflow dependency approach for modular building construction manufacturing process using Dependency Structure Matrix (DSM). *KSCE Journal of Civil Engineering*, *21*(5), 1525-1535. Doi: 10.1007/s12205-016-1085-1

Li, C. L., & Hall, N. G. (2019). Work package sizing and project performance. *Operations Research*, *67*(1), pp.123-142. Doi: 10.1287/opre.2018.1767

Li, D., & Lu, M. (2017). Automated generation of work breakdown structure and project network model for earthworks project planning: a flow network-based optimization approach. *Journal of Construction Engineering and Management*, *143*(1), 04016086. Doi: 10.1061/(ASCE)CO.1943-7862.0001214

Li, X., Shen, G. Q., Wu, P., Xue, F., Chi, H. L., & Li, C. Z. (2019a). Developing a conceptual framework of smart work packaging for constraints management in prefabrication housing production. *Advanced Engineering Informatics*, *42*, 100938. Doi: 10.1016/j.aei.2019.100938

Li, X., Wu, C., Wu, P., Xiang, L., Shen, G. Q., Vick, S., & Li, C. Z. (2019b). Smart work package-enabled constraints modeling for on-site assembly process of prefabrication housing production. *Journal of Cleaner Production*, *239*, 117991. Doi: 10.1016/j.jclepro.2019.117991

Lin, J. R., Hu, Z. Z., Li, J. L., & Chen, L. M. (2020). Understanding On-Site Inspection of Construction Projects Based on Keyword Extraction and Topic Modeling. *IEEE Access*, *8*, pp.198503-198517. Doi: 10.1109/ACCESS.2020.3035214

Liu, H., Al-Hussein, M., & Lu, M. (2015). BIM-based integrated approach for detailed construction scheduling under resource constraints. *Automation in Construction*, *53*, pp.29-43. Doi: 10.1016/j.autcon.2015.03.008

Liu, H., Lu, M., & Al-Hussein, M. (2016). Ontology-based semantic approach for construction-oriented quantity take-off from BIM models in the light-frame building industry. *Advanced Engineering Informatics*, *30*(2), pp.190-207. Doi: 10.1016/j.aei.2016.03.001

Marcus, A., & Maletic, J. I. (2003, May). Recovering documentation-to-source-code traceability links using latent semantic indexing. In *25th International Conference on Software Engineering, 2003. Proceedings.* (pp. 125-135). IEEE. Doi: 10.1109/ICSE.2003.1201194

Newman, D., Asuncion, A. U., Smyth, P., & Welling, M. (2007, December). Distributed inference for latent Dirichlet allocation. In *NIPS* (Vol. 20, pp. 1081-1088). Doi: 10.5555/2981562.2981698

Park, J., & Cai, H. (2017). WBS-based dynamic multi-dimensional BIM database for total construction as-built documentation. *Automation in Construction*, *77*, 15-23. Doi: 10.1016/j.autcon.2017.01.021

Pennington, J., Socher, R., & Manning, C. D. (2014, October). Glove: Global vectors for word representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)* (pp. 1532-1543). Doi: 10.3115/v1/D14-1162

Pessiot, J. F., Kim, Y. M., Amini, M. R., & Gallinari, P. (2010). Improving document clustering in a learned concept space. *Information Processing & Management*, *46*(2), pp.180-192. Doi: 10.1016/j.ipm.2009.09.007

Project Management Institute. (2021). Guide to the Project Management Body of Knowledge (PMBOK Guide). *Project Management Institute*, Newtown Square, PA, 7th ed. Retrieved from https://www.pmi.org/pmbok-guide-standards/foundational/PMBOK on 12th Sep

2021

Ramasesh, R. V., & Browning, T. R. (2014). A conceptual framework for tackling knowable unknown unknowns in project management. *Journal of Operations Management*, *32*(4), pp.190-204. Doi: 10.1016/j.jom.2014.03.003

Raz, T., & Globerson, S. (1998). Effective sizing and content definition of work packages. *Project Management Journal*, *29*(4), pp.17-23. Doi: 10.1177/875697289802900403

Salama, T., Salah, A., Moselhi, O., & Al-Hussein, M. (2017). Near optimum selection of module configuration for efficient modular construction. *Automation in Construction*, *83*, pp.316-329. Doi: 10.1016/j.autcon.2017.03.008

Siami-Irdemoosa, E., Dindarloo, S. R., & Sharifzadeh, M. (2015). Work breakdown structure (WBS) development for underground construction. *Automation in Construction*, *58*, pp.85-94. Doi: 10.1016/j.autcon.2015.07.016

Sutrisna, M., Ramanayaka, C. D., & Goulding, J. S. (2018). Developing work breakdown structure matrix for managing offsite construction projects. *Architectural Engineering and Design Management*, *14*(5), pp.381-397. Doi: 10.1080/17452007.2018.1477728

Teng, Y., Mao, C., Liu, G., & Wang, X. (2017). Analysis of stakeholder relationships in the industry chain of industrialized buildings in China. *Journal of Cleaner Production*, *152*, pp.387-398. Doi: 10.1016/j.jclepro.2017.03.094

Torkanfar, N., & Azar, E. R. (2020). Quantitative similarity assessment of construction projects using WBS-based metrics. *Advanced Engineering Informatics*, *46*, 101179. Doi: 10.1016/j.aei.2020.101179

Tsai, C. Y., & Chiu, C. C. (2008). Developing a feature weight self-adjustment mechanism for a k-means clustering algorithm. *Computational Statistics & Data Analysis*, *52*(10), pp.4658-4672. Doi: 10.1016/j.csda.2008.03.002

Wu, C., Wu, P., Wang, J., Jiang, R., Chen, M., & Wang, X. (2021a). Ontological knowledge base for concrete bridge rehabilitation project management. *Automation in Construction*, *121*, 103428. Doi: 10.1016/j.autcon.2020.103428

Wu, C., Wu, P., Wang, J., Jiang, R., Chen, M., & Wang, X. (2021b). Developing a hybrid approach to extract constraints-related information for constraint management. *Automation in Construction*, *124*, 103563. Doi: 10.1016/j.autcon.2021.103563

Wu, L., Ye, K., Gong, P., & Xing, J. (2019). Perceptions of governments towards mitigating the environmental impacts of expressway construction projects: A case of China. *Journal of Cleaner Production*, *236*, 117704. Doi: 10.1016/j.jclepro.2019.117704

Xu, Z., Zayed, T., & Niu, Y. (2020). Comparative analysis of modular construction practices in mainland China, Hong Kong and Singapore. *Journal of Cleaner Production*, *245*, 118861.

Yan, M., & Ye, K. (2007). Determining the number of clusters using the weighted gap statistic. *Biometrics*, *63*(4), pp.1031-1037. Doi: 10.1016/j.jclepro.2019.118861

Zhang, S., Boukamp, F., & Teizer, J. (2015). Ontology-based semantic modeling of construction safety knowledge: Towards automated safety planning for job hazard analysis (JHA). *Automation in Construction*, *52*, pp.29-41. Doi: 10.1016/j.autcon.2015.02.005

Zhong, B., Pan, X., Love, P. E., Sun, J., & Tao, C. (2020). Hazard analysis: a deep learning and text mining framework for accident prevention. *Advanced Engineering Informatics*, *46*, 101152. Doi: 10.1016/j.aei.2020.101152

Zhou, Z., Goh, Y. M., & Shen, L. (2016). Overview and analysis of ontology studies supporting the development of the construction industry. *Journal of Computing in Civil Engineering*, *30*(6), 04016026. Doi: 10.1061/(ASCE)CP.1943-5487.0000594

Zhu, A., Pauwels, P., & De Vries, B. (2021). Smart component-oriented method of construction robot coordination for prefabricated housing. *Automation in Construction*, *129*, 103778. Doi: 10.1016/j.autcon.2021.103778