https://dx.doi.org/10.1016/j.cie.2021.107816

A hypergraph-based approach for context-aware smart product-service system configuration

Zuoxu Wang[a,b], Chun-Hsien Chen[a], Pai Zheng[b,*] , Xinyu Li[c,**], Wenyan Song[d]

[a]*School of Mechanical and Aerospace Engineering, Nanyang Technological University, 639798*

[b]*Department of Industrial and Systems Engineering, Hong Kong Polytechnic University, Hung Hom, Hong Kong Special Administrative Region, 999077*

[c] *College of Mechanical Engineering, Donghua University, Shanghai 201620, China*

[d] *School of Economics and Management, Beihang University, 100191, Beijing, China*

*** Corresponding author:** pai.zheng@polyu.edu.hk

Full postal address: EF615, Core E, Hong Kong Polytechnic University, Kowloon, HKSAR, 999077

Informative title: A hypergraph-based Smart PSS configuration approach

Acknowledgement

The authors wish to acknowledge the funding support from the National Natural Science Foundation of China (No. 52005424), and Jiangsu Provincial Policy Guidance Program (Hong Kong/Macau/Taiwan Science and Technology Cooperation, BZ2020049).

Abstract

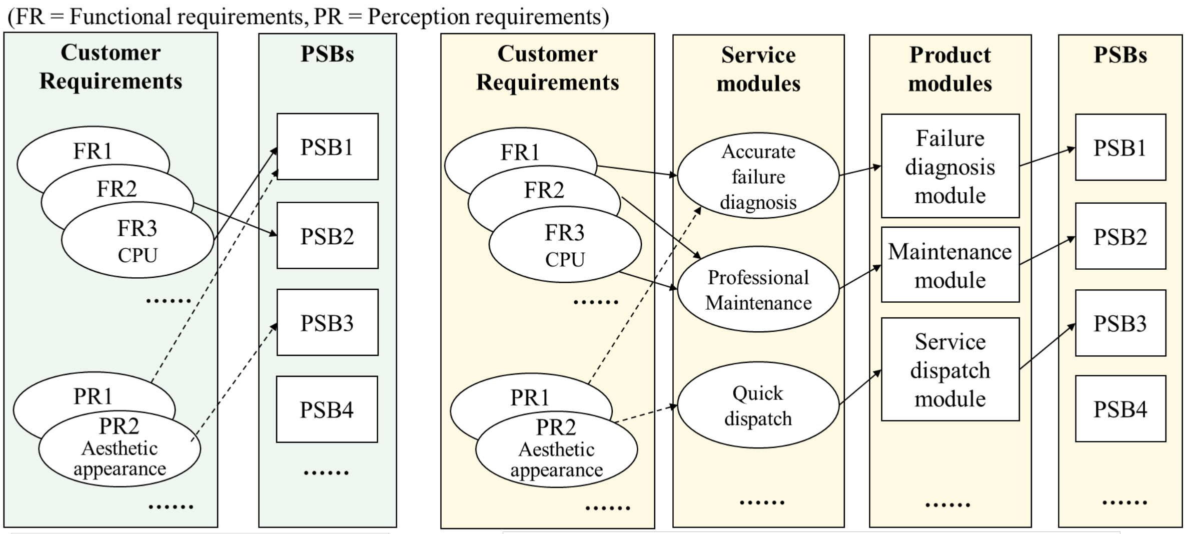
Smart product-service system (Smart PSS) is a heterogeneous and integrated system in which manufacturers/service providers deliver integrated and customized product-service bundles (PSBs) in a sustainable manner. Smart PSS configuration, a process of selecting proper PSBs based on user requirements, is expected to be further flexible to adjust the configuration results based on users' straightforward requirements that contain user-preferred usage scenarios and technical attributes. However, the technical attributes provided by users might be inaccurate due to a lack of professional knowledge, thus prone to unaligned configuration results. Besides, sorely technical attributes other than a comprehensive scope of information was emphasized in the conventional PSS configurators. To address these problems, this paper proposes a novel hypergraph-based Smart PSS configuration framework. Contrary to the conventional configurators that emphasize the mapping between technical attributes and PSBs, the proposed framework introduces usage scenario information as the auxiliary information, which is usually straightforwardly expressed by users. To represent the heterogeneous information with less local information loss, a hypergraph model based on the requirement attribute-product-service bundles-usage scenario (RA-PSB-US) data model is established. Besides, a hypergraph ranking algorithm is adapted to rank the PSBs. To mitigate the proneness of selecting settled PSBs other than returning adaptable results (i.e., the bias of hypergraph-based Smart PSS configuration) during the configuration process, an unbiased hypergraph ranking algorithm is proposed by normalizing the hyperedge degree. Finally, a comparative study is conducted based on an online 3D printing service review dataset to validate its effectiveness and advantages.

Keywords: Smart product-service systems; configuration system; hypergraph; context-aware; user-centric design

1 Introduction

With the trend of digital servitization, Smart product-service systems (Smart PSS) appeared as an integrated system for manufacturers to deliver user-required customized and integrated product-service bundles (PSBs) sustainably (Li et al., 2021b; Valencia Cardona et al., 2014; Zheng et al., 2018). By offering integrated PSBs, users can obtain customized functions, manufacturers/service providers can achieve higher customer loyalty and more profits through the prolonged product lifecycle. Many Smart PSS examples can be found in both B2C market and B2B market (Chang et al., 2018). In the B2C market, smart mobile phone is a typical example in which the physical phone and the apps collaborate as inseparable PSBs. The physical phone is the medium to offer Apps, and the Apps as the services offer the customized functions to the users. In the B2B market, one successful Smart PSS example is the 'power-by-the-hour' strategy from Rolls-Royce that Rolls-Royce offers flexible aeroplane engine usage plan together with customized maintenance services to extend the products' service life.

To fulfil user requirements with proper customized PSBs, configuration task is conducted as a decision-making process between user requirements and the available PSBs (Long et al., 2013; Yang et al., 2018). Conventional PSS configuration is a direct mapping between RAs and PSBs (Long et al., 2013) or a multilayer mapping among requirements, product modules and service flows (Song & Chan, 2015), as shown in Fig.1. Two limitations exist for the extant PSS configuration. On the one hand, well-defined technical attributes are required from users in the conventional PSS configuration systems. However, facing the much higher complexity of the smart and connected PSBs, users might offer inaccurate functional attributes due to the lack of professional domain knowledge (Dou et al., 2016; Wang et al., 2020) so that the conventional configurators are prone to the unaligned configuration results. For example, while configuring a desktop computer, the users usually cannot clearly identify the technical attributes, such as CPU, RAM, graphic card, but only express their requirements or comment on the current PSBs by straightforward natural language, e.g., '*I want a computer with Word/PowerPoint for my study* *in university*'. With inaccurate inputs, accurate outputs cannot be guaranteed in practice. On the other hand, solely technical attributes rather than a comprehensive scope of information are applied to configure PSBs (Pan et al., 2017; Yang et al., 2018), which will further reinforce the dependency on accurate technical attributes. To relieve the limitations, a more comprehensive configuration system is expected.



(a) Direct PSS configuration mapping

(b) Multi-layer PSS configuration mapping (derived from (Song and Chan 2015) [7])

Fig.1. Schematic diagrams of conventional PSS configuration

In Smart PSS, the configuration task has become a complex decision-making process with more information involved and is required to be flexible facing the individualized user requirements (Dou et al., 2020). Smart PSS has the unique features of high autonomy and value co-creation manner with intense user participation (Zheng et al., 2019b). Specifically, high autonomy is reflected in the capability of context-awareness and self-adaptability (Valencia Cardona et al., 2014; Wang et al., 2019b). Context-awareness was emphasized in a Smart PSS survey that the end-users expect customized rather than generalized product-service bundles (PSBs) (Valencia Cardona et al., 2014). A widely accepted definition of context is 'any information that can characterize the situation of an entity in the environment' (Dey, 2001). When configuring a PSB for end-users, the contexts refer to user feelings, expectations, or experiences instead of the development environment of Smart PSS (Chien et al., 2016). Another autonomy capability, self-adaptability, indicates that the PSS can adjust its performance according to the change of context or environment (Li et al., 2021a; Rijsdijk & Hultink, 2009).

Those Smart PSS features force Smart PSS configuration update in three aspects. First, to achieve context-awareness in Smart PSS, contextual information such as '*for novice users*' can be involved in Smart PSS configuration, which has strength in characterizing the user requirements and the PSBs (Shen et al., 2017). Although usage scenarios have been emphasized in the previous studies (Valencia Cardona et al., 2014), their influence on Smart PSS configuration has not been clarified yet. Second, given different user requirements, Smart PSS configuration needs to offer personalized PSBs other than 'off the shelf' ones (Dou et al., 2016; Shen et al., 2017), which reflects the capability of self-adaptability of Smart PSS configuration. Third, to quickly respond to the users' requests and further to achieve value co-creation manner in the early development stage and the usage stage (Zheng et al., 2019a), Smart PSS configuration process is also supposed to be conducted in an automatic way instead of a manual process (Dunke & Nickel, 2020). Under this situation, Smart PSS configurators, working as a decision-support tool for a complex system containing heterogeneous information, need to be upgraded as an IT-enabled design toolkit to return customized results beyond the product family scope (Lee & Kao, 2014).

Faced with the above challenges, this paper proposed a hypergraph-based Smart PSS configuration framework by simultaneously handling technical attributes and contextual information in requirements. In the proposed framework, usage scenarios, as a type of contextual information, are allowed for end-users to express their expected or encountered usage scenarios as supplementary information to alleviate the lack of domain knowledge. A hypergraph-based model is proposed to organize all the multi-sourced information in Smart PSS configuration for the first time. Hypergraph is applied in this paper due to its strength in handling heterogeneous entities and complicated relationships (Zhou et al., 2007). To adapt the hypergraph into the Smart PSS configuration, a hypergraph ranking algorithm is exploited. To further mitigate a practical configuration issue of returning settled PSBs rather than personalized PSBs in Smart PSS, which is caused by the unequal hyperedge degrees (i.e., the bias in hypergraph), an unbiased hypergraph ranking algorithm (UHR) is proposed by normalizing the hyperedge degrees.

The rest of this paper is structured as follows. The related works on PSS configuration and the preliminary on the hypergraph model are discussed in Section 2. In Section 3, the Smart PSS configuration framework, the hypergraph data model and schema, and a hypergraph-based ranking algorithm for Smart PSS configuration are expounded. An experiment based on an online 3D printing service platform is conducted as an example, and its results are discussed in Section 4. Finally, we summarized the scientific contributions and the future works in Section 5.

2 Literature review

To understand the state-of-the-art of the context-aware Smart PSS configuration models/approaches and to clarify the reason for deploying a graph-based model for Smart PSS configuration, this section summarizes the literature about context-aware Smart PSS (Section 2.1), PSS configuration (Section 2.2) and graph-based model in PSS/Smart PSS (Section 2.3). With the keywords '*product-service system configuration*', all the literature was sorted from the Web of Science (WoS) database owing to the broad coverage of the academic peer-reviewed articles. It is worthwhile noting that although plenty of articles discussed other factors (e.g., the uncertainty in the supply chain) in product configuration, they are not summarized in this paper since they are out of the scope of context-aware Smart PSS configuration. Additionally, the preliminary knowledge of the hypergraph is introduced in Section 2.4. Through the discussion on the existing literature, three research gaps are derived, as addressed in Section 2.5.

2.1 Context-aware Smart PSS

Context-aware systems, referring to the systems that use context to provide relevant information and service to the user, have become pervading in academia and industry (Alegre et al., 2016; Bolchini et al., 2007). Context-aware Smart PSS is described as the Smart PSS that can react and adapt its results based on different contextual information. In recent years, numerous applications, such as personalized recommendation (Champiri et al., 2015) and smartphone services (Chen et al., 2014) have been developed. To realize such a system, the context modelling and representation will be briefly discussed in this subsection, serving as the foundational overview of the proposed context-aware Smart PSS configuration framework.

In context-aware Smart PSS, context information can be collected via sensors or user interfaces. The context information collected via sensors is usually about the surrounding environment, such as time, location, temperature, etc. The one collected via user interfaces is more advanced, which could be users' activity logs or their feedbacks (Li et al., 2020). Typical context modelling techniques, such as key-value, logic-based, object-based, ontology-based, and graphical models, have already been discussed and surveyed in many studies (Alegre et al., 2016; Bolchini et al., 2007; Pradeep & Krishnamoorthy, 2019). Their strengths and limitations are briefly summarized in Table 1. To facilitate the subsequent hypergraph-based configuration process, key-value representation is applied due to its strength in simple representation and high compatibility with other information.

Table 1. Comparison of context representation models (derived from (Pradeep & Krishnamoorthy, 2019))

Context representation models	**Description**	**Strengths**	**Limitations**
Key-value	Describe context information as a list of context factors and their values	• Simple representation • Fast processing, storage, and lookup	• Limited capability of handling dynamics
Logic-based	Define context as rules Stored in the knowledge base	• Support logical reasoning	• Lack of standards
Object-based	Use objects to represent contexts	• Easy integration with high-level programming language	• Lack of reasoning • Lack of standards
Ontology-based	Represent context concepts with their relations and	• Enable dynamic relations among entities	• Limited capability of handling

	interdependent properties	• Higher expressiveness	uncertainty
Graphical	Use relationships modelling, e.g., Unified Modeling Language (UML) (Chen et al., 2014) and Context Dimension Tree (Bolchini et al., 2013; Curino et al., 2006; Javadian Sabet et al., 2020)	• Ease of information retrieval • Supports validation using constraints	• Complex querying

2.2 PSS configuration

The existing PSS configuration studies can be basically grouped as feature-based models, ontology-based models, and so on (Guillon et al., 2020).

On the one hand, the feature-based models formalize the PSS configuration process as either direct mapping (Long et al., 2013) or a multilayer mapping among requirements, product modules and service flow (Song & Chan, 2015). They have been widely accepted because of their strength in matching requirements to functional attributes. Roy et al. (2009) designed a features-based PSS configuration framework throughout the PSS lifecycle. This pioneering study defined the necessary steps of configuring PSS, such as identifying product/service structures and PSS lifecycle, identifying the variants' limitation, generating reasonable PSS variants and other associate steps. However, it still had the limitations that (1) aiming to satisfy the technical functions but omitting the customers' perceptions; and (2) analyzing the product components and the service modules singly. Mannweiler and Aurich (2011) then reinforced the PSS configuration framework by defining the sub-phrases of PSS configuration as requirement identification, configuration phase and purchasing phase, making their proposed framework an operational and standardized structure for the companies to follow. But it still stays at the functional level. Long et al. (2013) further extend the scope of the PSS configuration framework by introducing the customers' perceptional requirements. They used the support vector machine to automatically generate potential PSS configuration solutions. It was proved effective in improving user experiences with personalized PSBs, whereas the way of collecting customers' perceptional requirements was factor analysis, completed based on the engineers' knowledge. Moreover, Long et al. (2016) also boosted the PSS configuration research by rapidly extracting configuration rules based on a rough-set-based approach.

Recently, considering PSS's complex organizational structure with heterogeneous entities, including products, services, resources, and stakeholders, graph-based models (the so-called complex networks) were also applied in feature-based models for PSS configuration (Chang et al., 2018). For instance, Zhang et al. (2020) transformed the focus from the product-oriented configuration and emphasized service performance in PSS configuration. A graph-based model was proposed to present the interactions among PSS entities, and the corresponding algorithms were developed to derive the optimal solutions. Chang et al. (2018) also applied complex networks covering functions, product components, and services to configure the PSS's functions with higher availability. However, the studies of feature-based models on configuring PSBs for user requirements with the graph-based models are still insufficient and waiting for comprehensive PSS studies.

On the other hand, ontology-based approaches have also been applied to support PSS configuration. Shen et al. (2012) addressed that intensive and well-structured knowledge is necessary for the product extension services (PES) configuration. To organize the knowledge, they developed a configuration system based on three meta-ontologies, including service sub-ontology, product sub-ontology and customer sub-ontology. Wang et al. (2014) also built a meta-ontology of product-service systems, including five classes: Service Package, Function Modules, Service Flow, Process Modules, Service Elements, Service Resource, Interface, and Constraints. Then the configuration rules were expressed as general association rules to match the customer needs and the services together. Since the ontology-based methods highly depend on the comprehensive ontologies/knowledge bases as the prerequisite, meanwhile, they aim to promote knowledge reuse and sharing for the configurator instead of the end-users, the ontology-based methods are unsuitable for the Smart PSS configuration with the context-aware consideration for end-users.

As a result, based on the pioneers' achievements on PSS configuration, the contextual information can fundamentally serve as the auxiliary information to promote context-awareness for end-users and improve user-friendliness. Moreover, the PSS configuration methods involving factor extractions (e.g., the study of Long et al. (2013)) can be further enhanced by automatically extracting key phrase extraction techniques (Rose et al., 2010) other than highly depending on engineers' experiences/knowledge. In this way, the configuration approach in Smart PSS is hoped to be more context-aware, quickly reactive, and less domain knowledge required.

2.3 Ordinary graph in Smart PSS vs hypergraph in Smart PSS

Numerous extant PSS studies have shown the strength of graph models in representing heterogeneous entities (Ren et al., 2019). For example, Kim et al. (2009) utilized graphs and ontologies to represent PSS, in which the graph consists of values, products, services, and their relations. The PSS graph's effectiveness was proved by showing the complex usage scenarios of a meal assembly kitchen and its compatibility with the ontology. Wang et al. (2019a) established a graph covering product components, service modules, and contexts in Smart PSS to explore the latent relations among those entities. Li et al. (2014) applied a weighted complex network containing product components to simulate the complex product systems. Similarly, Sheng et al. (2015) constructed a directed graph for the CNC product-service system representation and adopted the Design Structure Matrix (DSM) method for modularization.

However, major PSS studies using graph models merely used the ordinary graph that the edges only represent pairwise relations. In practice, the relations between entities can be even more complicated, including one-to-many, many-to-one, or many-to-many relations. For example, as shown in Fig.2, it is assumed that a user ordered a PSB to meet several requirement attributes (RAs) with his/her expected usage scenarios (USs). This situation will be represented as {PSB, RA1}, {PSB, RA2}, {PSB, US1} and other edges based on an ordinary graph, making the relations between RAs and USs lost. When the number of entities increases in a PSS application, the local information loss will probably cause inaccurate PSB results. Nevertheless, hypergraph can integrate the multi-sourced information by using a set {PSB, RA1, RA2, US1, …} to represent the relationship without local information loss.

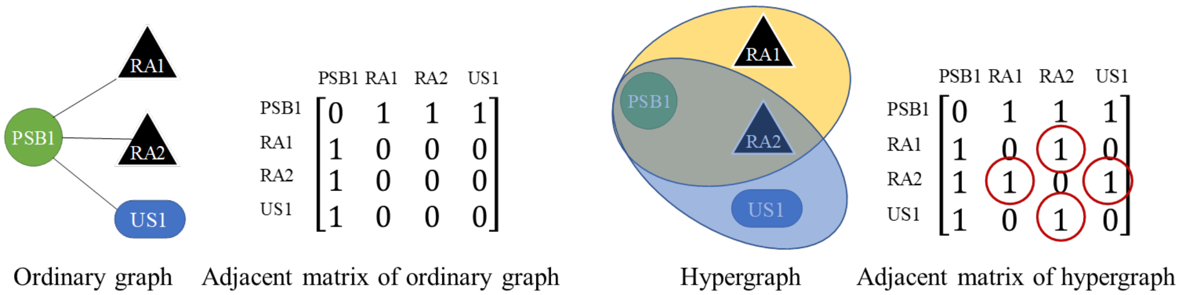


Fig.2. Comparison between ordinary graph and hypergraph.

2.4 Hypergraph model and the bias in hypergraph

A hypergraph is the generalization of a graph in which a hyperedge connects a finite number of nodes rather than two nodes (Zhou et al., 2007). The hypergraph models have been attempted in many applications, for instance, mass customization-oriented recommendations (Mao et al., 2019) and image classification (Yu et al., 2012). However, little research has attempted to deploy the Smart PSS configuration task on a hypergraph. When conducting a Smart PSS configuration task, various objects are involved, including products, services, and users. The relations among Smart PSS objects are either pairwise or high-order, which fits into the hypergraph's scope.

To be more specific, in a hypergraph $G = (V, E)$, the vertices are represented as a set $V = \{v_1, v_2, \ldots\}$ and the hyperedges are denoted as $E = \{e_1, e_2, e_3 \ldots\}$. As aforementioned, each hyperedge e_i is a set of nodes. The hypergraph will degrade to an ordinary graph when all the hyperedges only link two nodes.

To mathematically operate the hypergraph, some matrices of hypergraph should be defined. The incident matrix $H \in \mathbb{R}^{|V|\times|E|}$ with element $h(v_i, e_j)$ is defined that $h(v, e) = 1$ if a vertex v is in a hyperedge e, otherwise it equals to 0, denoted as:

$$h(v_i, e_j) = \begin{cases} 1, & if\ v_i \in e_j \\ 0, & \text{otherwise} \end{cases} \tag{1}$$

The degree of a hypergraph e is defined as $\delta(e) = \sum_{v\in V} h(v, e)$, meaning the node count in a hyperedge. The degree of a vertex v is $d(v) = \sum_{v\in e} w(e)h(v, e)$, referring to the number of hyperedges linked with the vertex. Let $D_v \in \mathbb{R}^{|V|\times|V|}$ denote the node degree matrix and let $D_e \in \mathbb{R}^{|E|\times|E|}$ denote the hyperedge degree matrix. Both are diagonal matrices whose diagonal elements are the nodes degrees and hyperedge degrees, respectively.

In Smart PSS configuration, a bias of selecting settled PSBs cannot be neglected. It is usually caused by the unequal hyperedge degrees $\delta(e)$ in the hypergraph (i.e., the bias in hypergraph) (Mao et al., 2019). For instance, PSBs are differentiated by their technical attributes (Chang et al., 2018). Some of them are high-end among the product-service family so they contain lots of technical attributes, making the hyperedge presenting their relations contains lots of entities. Hence the hyperedge degree will be high. Another example can also be seen that if a PSB fits in many usage scenarios, then the hyperedge containing the PSB and related usage scenarios will have a higher degree. The unbalances on the hyperedge happens because of the essential feature of the multipartite hypergraph, it causes the result that the PSBs with fewer technical attributes and usage scenarios tend to be selected, which cannot select personalized solutions for users in Smart PSS practical application. To mitigate the problem, it is necessary to normalize the hyperedge degrees, making the hypergraph unbiased.

2.5 Research gaps

To summarize, there is still a farther way to a Smart PSS configuration approach that has a comprehensive representation manner. Specifically, the research gaps mainly have three-folds. First, facing the complex organizational structure of Smart PSS, although the trend of graph-based models has already emerged, a comprehensive graph-based model containing contextual information is still a preliminary study. Second, derived from previous graph-based models in PSS representation and modularization, an ordinary graph illustrating pairwise relations is still leveraged in some tentative studies of PSS configuration, which actually neglects some essential information and hence drives the configuration ranking algorithm to select some inappropriate PSBs. Finally, despite the advantages of hypergraph, the hyperedge bias on the existing hypergraph is scarcely addressed, which is critical in the configuration process. To overcome these bottlenecks, a novel approach for context-aware Smart PSS configuration is significantly essential.

3 Methodology

Fig. 3 depicts the hypergraph-based Smart PSS configuration framework, which contains four levels in terms of the system architecture, namely, (1) data resources, (2) data model, (3) hypergraph construction, and (4) Smart PSS configuration application level. Particularly, Level (1)(2)(3) are conducted offline. They reflect the context-awareness of the proposed framework by defining usage scenario (US) information as one kind of significant entities in the Smart PSS configuration process. The Smart PSS configuration application level (Level (4)) undertakes the user query process online. Unlike the conventional PSS configuration framework that focuses on mapping the solutions with the technical attributes, the proposed Smart PSS configuration framework allows the user to express their expected usage scenarios and experiences during the configuration process. Given different user requests containing US information, the configuration framework can return corresponding PSB ranks, thus reflecting the context-awareness and, to some extent, the self-adaptability. A novel UHR algorithm is also developed at the application level to achieve the context-awareness of the proposed framework.

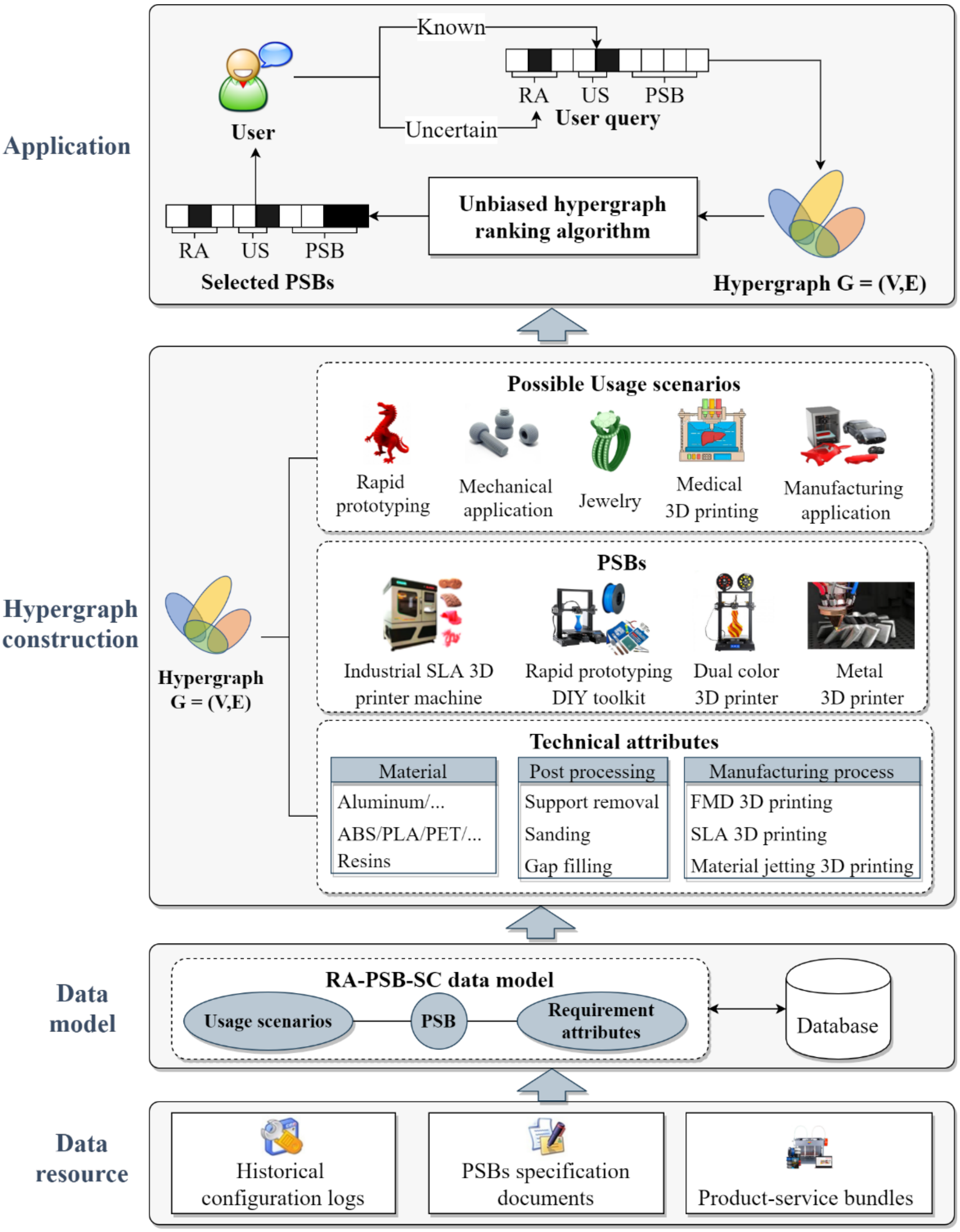


Fig. 3. Overview of the hypergraph-based Smart PSS configuration framework

3.1 RA-PSB-US data model

The development team needs to collect that information in order to build up a hypergraph model and then train the algorithm based on the accessible data. In Smart PSS, both the RAs and the USs are accessible. The technical RAs refers to the function parameters of the PSBs, such as weight, model size, brand, material, and so on. These RAs can be identified according to the functional specification documents of the PSBs. Meanwhile, the usage scenarios indicate the users' feelings/perceptions/experiences in the usage phase, such as good quality, fast delivery, easy installation, etc. Note that the usage scenario information can also be regarded as the perceptional requirements mentioned in Section 2 's literature summary. They can be formalized as key phrases and extracted from the user reviews. Furthermore, the PSBs are the customized solutions offered by the service providers or manufacturers.

Hence, there are three kinds of information in the Smart PSS configuration. Two relations can intuitively link them, namely, the relations between the PSBs and the RAs from the functionality specification documents (i.e., **Product-service bundle-Requirements attributes relationship (PSB-RA)**), and the relations between the PSBs and the USs from the user reviews (**Product-service bundle-Usage scenario relationship (PSB-US)**). Fig. 4 shows the schema of the proposed RA-PSB-US data model. The relations are explained as follows.

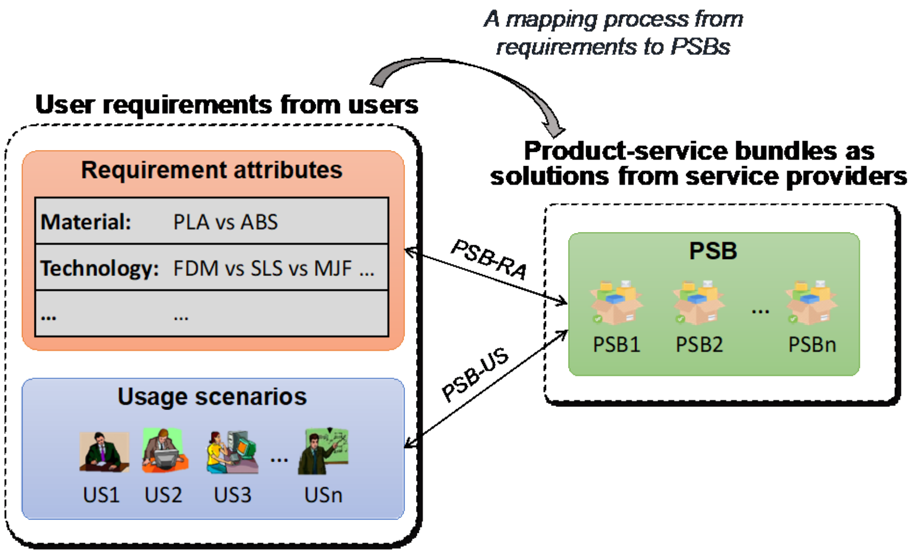


Fig. 4. Objects and relations in the Smart PSS configuration

To identify PSB-RA relations, a direct mapping is feasible to automatically complete this task when the functional parameters are stored in the structured tables. In the functional specification documents, each PSB is usually configured by the values on different functional parameters, and hence it can be linked with many RAs. The keyword/key phrase extraction techniques can also be adopted for some documents that are written with natural language.

A PSB can also be connected with many USs according to user reviews and user-related information. Among the user reviews, the details about the usage scenarios such as '*using at home*' can be extracted and then be linked with the corresponding PSB. Since the user reviews are always expressed by natural language, the key phrase extraction techniques should be applied to identify the PSB-US relations. Moreover, the user-related information collected via the PSS configuration platform can also link the PSBs with potential contexts. For example, if the user's historical configuration orders and location are collected, then the PSBs can be connected with the location information.

Although the involved entities might have other relations, e.g., the relationship between USs and RAs, those relations will hardly be available unless depending on the experts' professional knowledge or empirical experiences. For instance, given a specific usage scenario '*for a delicate jewellery*' when configuring a 3D printing service, although it can be linked with several RAs, such as '*Stereolithography* or *multi-jet moulding technology'* as the printing technology and '*water-soluble consumables'* as the material, these relations will be hard to get from users and will be costly to get from the experts.

3.2 Hypergraph construction

To build the hypergraph, the involved information, including PSBs, RAs, and USs, are treated as nodes and denoted as $PSB = \{psb_1, psb_2, psb_3, \ldots\}$, $US = \{us_1, us_2, us_3, \ldots\}$, and $RA = \{ra_1, ra_2, ra_3, \ldots\}$. Their union set constitutes the node-set V for the hypergraph $G = (V, E)$ that $V = PSB \cup US \cup RA$.

Based on the RA-PSB-US data model, the hyperedge E is defined by two subsets, namely $E^{(1)}$: the PSB-RA relation and $E^{(2)}$: the PSB-US relation, as shown in Table 2. Each hyperedge in $E^{(1)}$ will contain a PSB together with several RAs. Each hyperedge in $E^{(2)}$ is composed of a PSB and several USs, if the USs were mentioned in a piece of user review. Since we assume that each function specification document and each user review are equally important to the hypergraph, all the hyperedge weights are set as 1.

Table 2. Relations in the hypergraph model for Smart PSS configuration

Hyperedge No.	**Relation notation**	**Edge weight $\boldsymbol{w(e)}$**
$E^{(1)}$	RA-PSB	1
$E^{(2)}$	SC-PSB	1

A simple hypergraph example based on the RA-PSB-US data model is shown in Fig. 5, straightforwardly representing a simple Smart PSS configuration activity. The hypergraph contains four PSBs, seven RAs, and five usage scenarios. Totally eight hyperedges appear in the hypergraph, as shown below.

$$E = \{$$

$$'PSB-RA1':('RA1','RA2','RA3','PSB1'),$$

$$'PSB-RA2':('RA3','RA4','PSB2'),$$

$$'PSB-RA3':('RA2','RA3','RA5','RA6','RA7','PSB3'),$$

$$'PSB-RA4':('RA3',' RA4','PSB4'),$$

$$'PSB-US1':('US1','PSB1'),$$

$$'PSB-US2':('US3','US4','US5','PSB2'),$$

$$'PSB-US3':('US1','US2','US3','US4','PSB3'),$$

$$'PSB-US4':('US2','US4','PSB4')$$

$$\}.$$

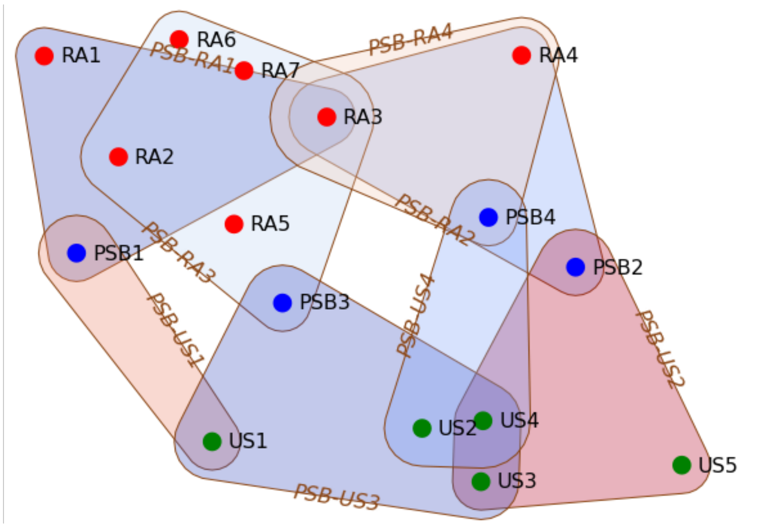


Fig. 5. A hypergraph example based on the RA-PSB-US data model

3.3 Unbiased hypergraph ranking algorithm

Essentially, Smart PSS configuration is a ranking problem to learn a scoring function given RAs and USs that returns sorted PSBs as outputs (Long et al., 2013). Specifically, when a user offers specific RAs or USs in the user queries, the system will sort the PSBs based on their ranking scores corresponding to the user queries. This problem can be solved by graph-based ranking. Mathematically, we aim to learn a ranking score vector $\mathbf{f}: V \rightarrow \mathbb{R}$ based on the hypergraph $G = (V, E)$ and the query vector $\boldsymbol{y} = [y_1, y_2, \dots, y_{|V|}]^T$, where y_i denotes the initial scores of nodes (Tan et al., 2011).

To keep the minimal lost for the ranking function, a cost function is defined based on the previous study of hypergraph ranking (Mao et al., 2019; Tan et al., 2011; Zhou et al., 2007).

$$Q(\mathbf{f}) = \frac{1}{2}\sum_{i,j=1}^{|V|}\sum_{e\in E}\frac{w(e)h(v_i,e)h(v_j,e)}{\delta(e)}\left\|\frac{\mathbf{f}_i}{\sqrt{d(v_i)}} - \frac{\mathbf{f}_j}{\sqrt{d(v_j)}}\right\|^2 + \mu\sum_{i=1}^{|V|}\|\mathbf{f}_i - y_i\|^2 \qquad (2)$$

, where $\mu > 0$ is the regularization parameter. The cost function $Q(\mathbf{f})$ cumulates the changes of the scoring vector between nearby nodes over the hyperedges on the hypergraph. The optimal solution of $\mathbf{f}^*$ is the one with the minimal cost $Q(f)$, $\mathbf{f}^*$can be derived by letting the gradient of $Q(\mathbf{f})$ as 0:

$$\left.\frac{\partial Q}{\partial \mathbf{f}}\right|_{\mathbf{f}=\mathbf{f}^*} = (\mathbf{I} - \mathbf{A})\mathbf{f}^* + \mu(\mathbf{f}^* - \boldsymbol{y}) = 0 \qquad (3)$$

The optimal ranking function $\mathbf{f}^*$ is deduced after solving Eq. (3), as shown in Eq. (4).

$$\mathbf{f}^* = (\mathbf{I} - \alpha\mathbf{A})^{-1}\boldsymbol{y} \tag{4}$$

, where $\alpha = 1/(\mu + 1) \in (0,1)$ is a parameter and $\mathbf{A} = \mathbf{D_v}^{-1/2}\mathbf{HWD_e}^{-1}\mathbf{H}^{\mathrm{T}}\mathbf{D_v}^{-1/2}$ is an intermediate matrix.

Except for the direct derivation method, the hypergraph ranking problem can also be resolved by the random walk process on the hypergraph (Mao et al., 2019). Given several starting nodes from the vertex set V, represented as a vector $\mathbf{q} \in \mathbb{R}^{|V|}$, in which the starting nodes equal to 1 and the others equal to 0, the random walk with restart model will transit to their adjacent vertices following edges with probability α or restart from the starting nodes with probability $(1 - \alpha)$. The starting nodes can be randomly selected during the training process. Let $\mathbf{p}^{(t)}$ be the vector whose elements are the transition probability from a node to other nodes at time t. Let $\mathbf{T} \in \mathbb{R}^{|V|\times|V|}$ be the transition matrix. The random walk on the graph is a recursive process that follows the equation:

$$\mathbf{p}^{(t+1)} = \alpha\mathbf{T}\mathbf{p}^{(t)} + (1 - \alpha)\mathbf{q} \tag{5}$$

The convergence of $\mathbf{p}^{(t)}$ will happen when $\mathbf{p}^{(c)} = \mathbf{p}^{(t+1)} = \mathbf{p}^{(t)}$ is satisfied. At this moment,

$$\mathbf{p}^{(c)} = (\mathbf{I} - \alpha\mathbf{T})^{-1}\mathbf{q} \tag{6}$$

, where $\mathbf{T} = \mathbf{D}_v^{-1}\mathbf{HWD}_e^{-1}\mathbf{H}^{\mathrm{T}}$. It is clear that $\mathbf{T}$ has a similar structure with matrix $\mathbf{A}$, and Eq. (6) has the same structure as Eq. (4). The equivalence has been proved in many research (Mao et al., 2019; Tan et al., 2011; Zhou et al., 2007). Here, $\mathbf{T}$ can be regarded as the normalization of matrix $\mathbf{A}$.

Hence, the scoring function $\mathbf{f}$ can be learned through the convergence process of random walks on a hypergraph. Nevertheless, the biases on the hypergraph force us to adapt the original hypergraph ranking algorithm. Because of the diversity on the PSBs' technical RAs and the different amount of information in the user comments, the hyperedges' degree δ_e will significantly vary, causing the bias on the hypergraph. The bias will make each transition lean towards the edges that have fewer nodes, which can be seen by the transition probability from vertex u to its adjacent vertex v:

$$p(u,v) = \sum_{e \in E} w(e) \frac{h(v,e)}{\delta(e)} \frac{h(u,e)}{d(u)} \tag{7}$$

As shown in Eq. (7), the transition probability is decided by three parts: the edge weight $w(e)$, the node degree $d(u)$ and the hyperedge degree $\delta(e)$. The smaller $\delta(e)$, the greater the $p(u,v)$. Eq (7) implies that given a node u, the random walk will firstly select a hyperedge e linked with node u, then select an adjacent node v among the selected hyperedge e. To solve the bias, the hyperedge degree should be averaged, where the adapted transition probability is defined as:

$$p(u,v)' = \sum_{e \in E} w(e) \frac{h(u,e)h(v,e)}{d(u)} \frac{\sqrt{\delta(e)}}{\sum_{e \in E} \sqrt{\delta(e)}} \tag{8}$$

Correspondingly, the modified transition matrix $\mathbf{T}'$ is derived $\mathbf{D}_{\mathrm{v}}^{-1}\mathbf{H}\mathbf{W}\mathbf{D}_{\mathrm{e}}^{\frac{1}{2}}(\mathbf{M}^{\mathrm{T}}\mathbf{D}_{\mathrm{e}}^{\frac{1}{2}}\mathbf{M})^{-1}\mathbf{H}^{\mathrm{T}}$, where $\mathbf{M} \in \mathbb{R}^{|\mathbf{V}|}$ is an associate column vector that all the elements equal to 1.

3.4 Multi-objective user request identification for Smart PSS configuration

Based on the UHR algorithm, the established hypergraph and the corresponding transition matrix for Smart PSS configuration can be trained offline, the next step is to identify the user requests as online inputs and accordingly rank the PSBs.

A raw user request in this study contains two parts, namely the selected RAs and the USs. Technically, the US information can be either manually provided by the users or automatically collected by the PSS system. On the one hand, the US information can be the textual information in user request, such as '*good quality*' and '*DIY for kid*'. Key phrase extraction techniques, e.g., TextRank (Mihalcea & Tarau; Zhang et al., 2018), collaborating with some contextual information templates can achieve the contextual information extraction from the user-supplied USs. The extracted key phrases are supposed to be matched to the most similar usage scenario once triggered by the similarity threshold. On the other hand, the US information can also be the user-related information collected via the PSS configuration platform, e.g., location. All the extracted US phrases are concatenated with the selected RAs as a vector, serving as the user request. Mathematically, the user requests containing both the RAs and/or USs can be represented as a query vector $\mathbf{q} \in \mathbb{R}^{|\boldsymbol{V}|}$, where $q_i = 1$ if the ith node on hypergraph is selected in the query vector, otherwise $q_i = 0$.

Except for the US information, the user queries can be further enriched by introducing the adjacent information of the user-selected nodes since they may reflect the user-preferred attributes. Specifically, the elements in initial query vector $\mathbf{q}$ are still set as 1 if they are mentioned by the user. Note that there is no need to ask the user to configure all the RAs and USs. The transit matrix $\mathbf{T}_{\mathrm{u,v}}$ is introduced, it presents the relatedness between u and v. Then the final query vector will be $\mathbf{y} = \mathbf{Tq}$. Based on the final query vector $\mathbf{y}$ and the trained ranking function $\mathbf{f}$, the top K PSBs with the highest-ranking scores can be selected and recommended to the users.

4 An illustrative example

To make the proposed hypergraph-based Smart PSS configuration framework consolidate, an illustrative example of online 3D printing services is shown to demonstrate the configuration process in Smart PSS.

Massive manufacturing companies have launched online 3D printing services as customized solutions to users, including the functions of instant price quotes, 3D model download, remote 3D printing, post-processing, delivery/shipping, etc. It is a typical PSS (Tukker, 2004) since it integrates both product and services, meanwhile involves multiple stakeholders' participation (i.e., user participation and service provider's contributions) via an online platform. However, there is still no way to achieve good context-awareness on the end-user side, let alone the quick adjustment to the customized PSS configuration results based on the contextual information. In this example, the online 3D printing platform can access closer to a Smart PSS on the end-user side by applying the proposed hypergraph-based Smart PSS configuration framework to derive automatic and customized configuration results with context-aware information.

The example simulates that the Smart PSS configuration platform collects user's preferred RAs, their expected USs, and the other automatically collected US information as the raw inputs. The system will return a list of top K PSBs regarding the ranking score to fulfil the user query. The parameter K indicates that how many PSBs will be selected in the result list.

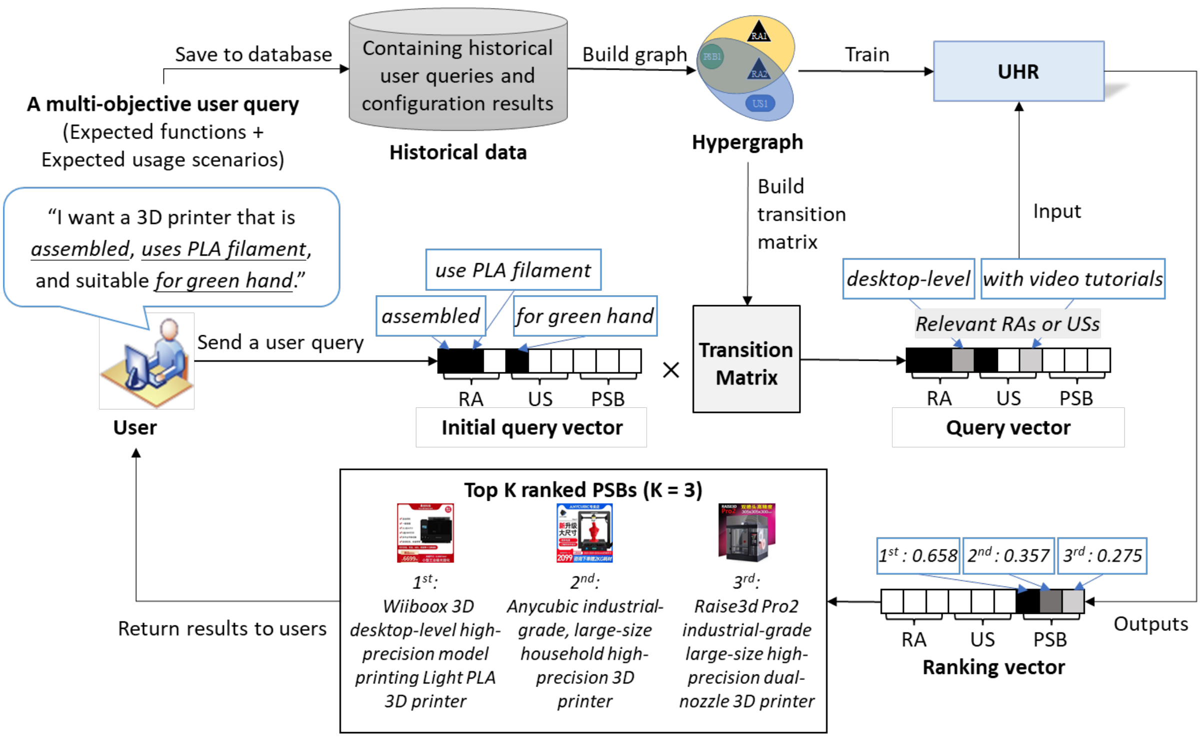


Fig. 6. Illustration example of the hypergraph-based configuration process

Fig. 6 shows the illustrative process of the proposed hypergraph-based configuration framework. Basically, the deployment of this framework consists of four steps, including *hypergraph construction*, *UHR model training process*, *user query generation*, and *PSB ranking*.

Step 1: Historical data collection and hypergraph construction. Beginning from the database in Fig. 6, a hypergraph will be established using the historical user queries and configuration logs. For instance, a pair of configuration log and user query could be stored based on key-value representation way: *{Assembled: Yes; Material: PLA; User context: for green hand}* and *{Name: Anycubic Cube Mega-x Large Size 3D Printer; Material: PLA, ABS; Size: 500x500x553mm; Number of nozzles: 1; Classification category: quasi-industrial grade; Assembled: Yes; Interface type: USB memory card; Brand: Anycubic; Time to market: 2019-12-19; Application areas:*

clothing, shoes, hats, education, scientific research, jewelry, accessories, biomedical, cultural, broadcasting, art houses; Printing speed: 20-100mm/s}. They are organized with the format of $E^{(1)}$: PSB-RA and $E^{(2)}$: PSB-US, respectively. The hypergraph could be established following the proposed RA-PSB-US data model.

Step 2: UHR model training process based on historical data. The UHR algorithm will train the intermediate matrices of the hypergraph, such as the transition matrix $\mathbf{T}'$**.** After the matrix T' is convergent, the ranking function **f** could be learned for PSB ranking.

Step 3: User query generation. In this step, the users could generate their user queries by indicating their expected functions and usage scenarios. The user queries, on the one hand, will be stored into the database for the hypergraph updation and the UHR model training in the future. On the other hand, each initial user query will be pre-processed by aggregating some relevant information according to $\mathbf{T}'$. For example, as shown in Fig. 6, additional information: "desktop-level" and "with video tutorials" will be aggregated into the user query vector by multiply $\mathbf{T}'$. Then the pre-processed user query will serve as the input of the ranking function **f.**

Step 4: PSB ranking based on user query and the trained UHR. Finally, the ranking function **f** will only return the probabilities of each PSB being selected (e.g., 1st: 0.658, 2nd: 0.357, 3rd:0.275) according to the pre-processed user query. The top K (K = 3 in the illustrative example of Fig. 6) PSB candidates will return to the users and be saved into the database.

5 Experiment

To validate the proposed hypergraph-based model, the performance of the proposed algorithm is also evaluated.

5.1 Data Collection and hypergraph construction

Firstly, a dataset containing the online 3D printing service bundles (i.e., the PSBs), the functionality specifications, and the user reviews about their experience after placing orders was collected. A total of 28 PSBs' functionality specification documents are collected, as listed in Table 3. From the documents, 219 RAs on 24 kinds of parameters are extracted, including material, nozzle number, category, manufacturer, printing speed, printing size, additional features, etc., as shown in Fig. 7. Besides, 2947 USs were extracted from 1714 pieces of user reviews using a Python Package, i.e., TextRank (Mihalcea & Tarau; Zhang et al., 2018). TextRank was chosen since it can be extended for phrases and short sentences extraction. Some extracted phrases with the same/similar meanings are integrated to keep the hypergraph concise. The typical and frequently-mentioned usage scenarios contain '*good quality*', '*quick delivery*', '*cost-effective*', '*free-damage packaging*', '*good customer service*', '*smooth surface*', '*for jewellery*', '*for creative design/DIY*', '*for rapid prototyping*', '*for automobile*', etc.

Table 3. Typical PSBs of the online 3D printing service

PSB id	**PSB name**
psb0	Raise3d Pro2 industrial-grade large-size high-precision dual-nozzle 3D printer
psb1	Raise3d Pro2 Plus dual-nozzle dual-color industrial-grade large-scale high-precision FDM 3D printer
psb2	Creative 3D LD-002R high-precision LCD large-size photosensitive resin desktop-level SLA 3D printer
psb3	Chuangxiang 3D ENDER-3S pro v2 high-precision quasi-industrial household 3D printer
……	……
psb24	Three green S8S desktop-level high-precision large-size printing quasi-industrial-grade FDM printing 3d printer
psb25	Aurora Ervo 3d printer Z-603S industrial-grade stable high-precision printing model home large-size 3D printer
psb26	Anycubic Mega-x Industrial-Grade 3D Printer Large-size Household High-precision Machine Z-axis Double Screw
psb27	Formlabs Form3 high-precision SLA industrial-grade ABS/plastic/photosensitive resin high temperature

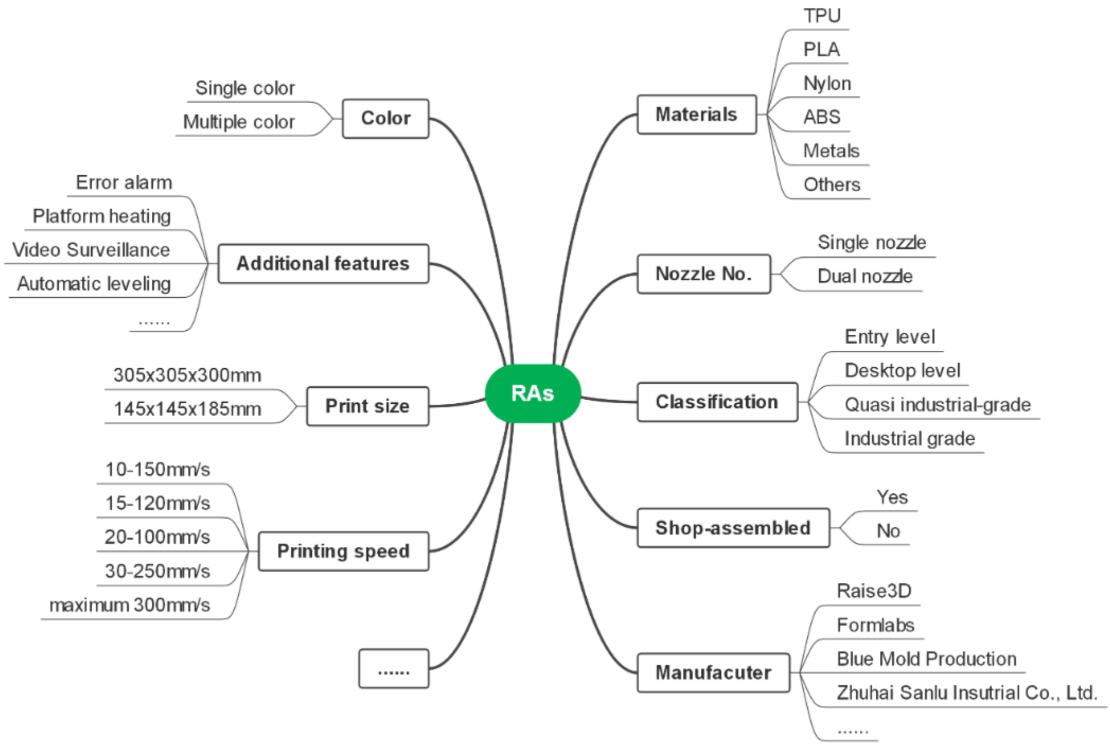


Fig. 7. RAs of the online 3D printing service

Based on the dataset, a hypergraph for the online 3D printing service configuration was built up. According to the hypergraph's statistical information in Table 4, it is clear to see a considerable difference between the edge sizes of $E^{(1)}$ and $E^{(2)}$. Especially noteworthy, the maximal hyperedge degree is 20 coming from $E^{(1)}$, whereas the minimal edge degree is 2, which belongs to $E^{(2)}$. The average edge degree is 3.9749. This phenomenon proves that the hyperedge degrees vary greatly, consistent with the hypergraph's bias in Section 3.3.

Table 4. Statistics of the hypergraph on 3D printing services

Element	Element name	Size
PSB	Product-service bundle	28
RA	Requirement attribute	219
US	Usage scenario	2947
$E^{(1)}$: PSB-RA	Product-service bundle-Requirement attribute relationship	28
$E^{(2)}$: PSB-US	Product-service bundle-Usage scenario relationship	1714

5.2 *Evaluation metrics and the compared graph-based models*

In this experiment, a total of 228 user reviews were chosen as the test dataset. Partial RAs and USs were hidden to simulate that the users only care about partial rather than all the PSB features.

To evaluate the performance, several evaluation metrics were selected and applied, including precision, recall, F1-score, mean average precision (MAP) and normalized discounted cumulative gain (NDCG), were applied. Specifically, precision refers to the number of correctly recommended PSBs over the total number of recommended PSBs. Recall is denoted as the number of truly and correctly recommended PSBs over the total number of PSBs that should be recommended. F1-score is the weighted average of precision and recall, which can be calculated via:

$$F_1 = \frac{2 \times Precision \times Recall}{Precision + Recall} \tag{9}$$

MAP can be calculated through the following equations.

$$\mathrm{P@K}(\pi, \mathrm{l}) = \left(\sum\nolimits_{t \le k} I\left\{l_{\pi^{-1}_{(t)}} = 1\right\}\right)/k \tag{10}$$

$$\mathrm{AP}(\pi, \mathrm{l}) = \left(\sum\nolimits_{k=1}^{K} P@k \times I\{l_{\pi^{-1}_{(t)}} = 1\}\right)/k_1 \tag{11}$$

Here, π refers to the ranking result of the PSB list. *I* is the indicator function. $l_{\pi^{-1}_{(t)}}$ means the label of the t-th PSB in the result list, in which if t-th PSB does belong to the user truly selected results, then its label equals 1, otherwise equals 0. $\mathrm{P@K}(\pi, \mathrm{l})$ indicates the ratio of the number of correctly recommended PSBs and the parameter K. In Equation (10), K means the number of PSBs returned in the result list, k_1 refers to the number of PSBs related to the query (i.e., the number of PSBs expected to be recommended). MAP is the mean of the AP values of all the queries.

NDCG is also a frequently used metric for ranking quality measurement. It can be calculated by defining

$$\mathrm{DCG} = \sum\nolimits_{k=1}^{t} rel_k / \log{(k+1)} \tag{12}$$

, where rel_k is the graded relevance of the result at the position k. In the example of the 3D printing service, rel_k can be defined as 1 if the result at position k is related to the queries, otherwise defined as 0. Then NDCG equal to

$$\mathrm{NDCG} = \mathrm{DCG}/\mathrm{IDCG} \tag{13}$$

, where IDCG refers to the ideal discounted cumulated gain.

The parameter K that indicates the number of selected PSBs is set as 2, 3, 5 or 8 since the total PSBs count is only 28 PSBs in this example. Larger K does not comfort the practical application because it will be easy to select truly recommended PSBs that cannot prove the algorithms' effectiveness.

Five algorithms are compared with the proposed approach, including a non-personalized approach (denoted as AVG), item-based collaborative filtering (ICF) (Sarwar et al., 2001), singular vector decomposition (SVD) (Brand, 2003) and two most related graph models. AVG ranks PSBs based on their average ratings. ICF and SVD, frequently used as recommendation approaches, rank the PSBs based on historical PSB ratings. AVG, ICF and SVD are non-context-aware models in this experiment since they are based on historical ratings rather than contextual information. A classic graph-based ranking model called PageRank (PR) was selected because it also uses the random walk model to learn the ranking scores. Its significant difference with hypergraph ranking lies in that PageRank is based on the ordinary graph. At the same time, the original hypergraph ranking (HR) algorithm was also deployed to compare with the UHR algorithm.

5.3 Comparison of the algorithm performance

The results of compared algorithms are displayed in Table 5. The first six rows show the algorithms' performance on handling the heterogeneous data simultaneously based on MAP, NDCG and F1-score. The results show that the UHR algorithm has the highest MAP value of **0.431**, followed by the HR's MAP value of **0.392**. Besides, the UHR also has the highest NDCG and F1-score values whenever n equals 2, 3, 5, or 8. It proves that the hypergraph model does exceed the ordinary model. Less information loss on the graph matrices leads to performance improvement on the ranking algorithm. Furthermore, normalizing the hypergraph degrees strengthens the original hypergraph ranking with better performance.

Table 5. Comparison with models

	MAP	ndcg@2	ndcg@3	ndcg@5	ndcg@8	F1@2	F1@3	F1@5	F1@8
AVG	0.150	0.071	0.091	0.127	0.172	0.805	0.805	0.714	0.317
ICF	0.213	0.106	0.135	0.177	0.232	0.850	0.822	0.764	0.678
SVD	0.210	0.119	0.142	0.185	0.225	0.852	0.824	0.766	0.674
PR	0.371	0.316	0.321	0.349	0.399	0.894	0.862	0.799	0.708
HR	0.392	0.344	0.335	0.378	0.427	0.896	0.862	0.802	0.711
UHR	**0.431**	**0.357**	**0.394**	**0.447**	0.494	**0.900**	**0.875**	**0.817**	**0.726**
UHR_RA	0.419	0.323	0.381	0.446	**0.505**	0.875	0.856	0.804	0.720

(UHR_RA = UHR based on only RAs without USs)

Fig. 8 represents three algorithms' precision-recall curve that the UHR algorithm still has the best performance.

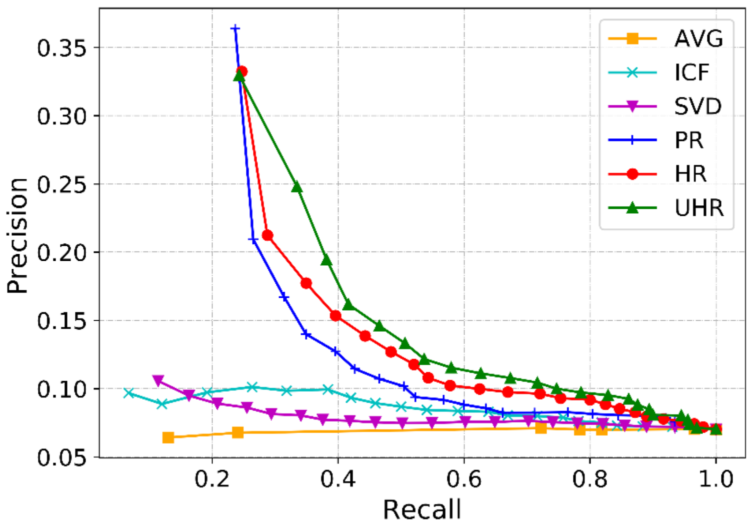


Fig. 8. Precision- recall curve

The stability of the ranking algorithms is tested as well. Initially, the parameter α was set as 0.9 when we compared the algorithms' performance based on the test set, meaning that during the iteration process of the random walk, a node will jump to its neighbor node with a probability of 0.9 and restart from the starting nodes with the probability of 0.1. To check the stability of the tested algorithms, the parameter α is gained from 0.60 to 0.95 with a step size of 0.05 and finally set as 0.999. The value of parameter α starts from 0.60 rather than an even smaller value because we hope the random walk process can transverse all the nodes on the hypergraph in the recursive process. Fig. 9 shows the trends of MAP and NDCG values of different algorithms when the parameter α changes. It is clearly seen that in a wide range of α, the UHR outperforms the PageRank and hypergraph ranking. MAP of the UHR is stable at around 0.43, and the value of NDCG is around 0.45.

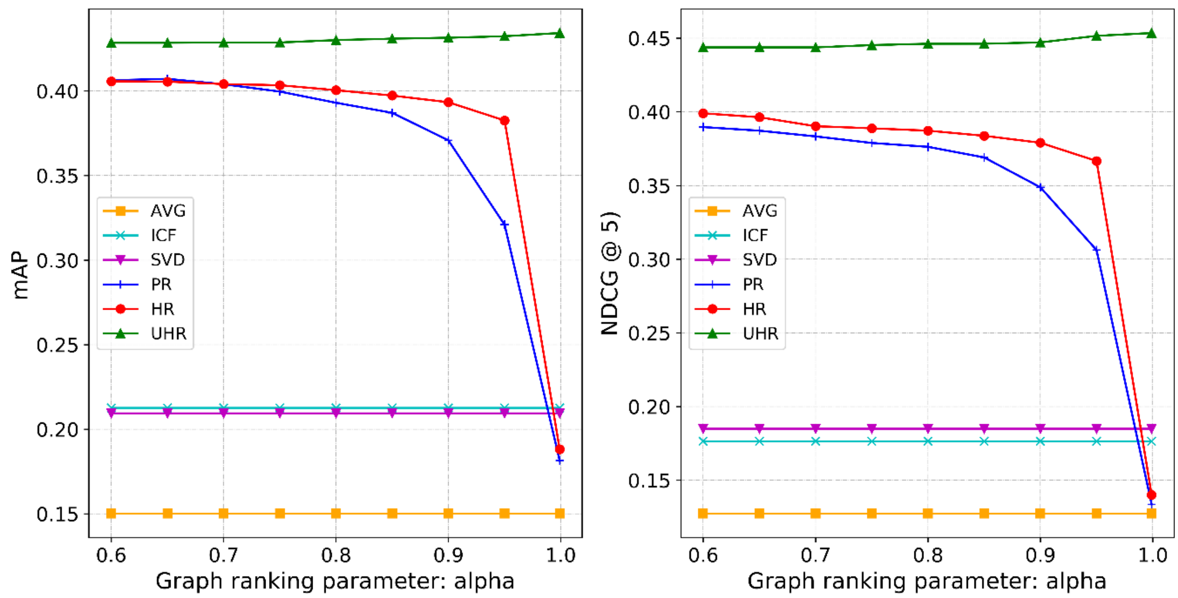


Fig. 9. Evaluation metrics under different alpha

5.4 Comparison of the effect of usage scenarios in Smart PSS configuration

To test the effect of the usage scenarios in Smart PSS configuration, user queries without USs and the user queries with both RAs and USs are utilized as inputs. The last two rows of Table 5 illustrate that the performance of the UHR was improved by adding US information. The improvement ratio is around 10% when the number of selected PSBs is 2. In other words, while selecting fewer PSBs, usage scenarios will play a more prominent role in the UHR. This phenomenon conforms to the industrial solution configuration application that although there are lots of solution options among all the industries, the proper alternatives for each scenario are limited so that the configuration list should be short and precise.

The performance improvements of introducing USs can also be seen from the query examples in Table 6. The PSBs in the second column are the ground-truth results from historical configuration logs that the users finally selected. Under various user requests, it is clear that the results using both RAs and USs can rank the ground-truth results as the top-ranking candidates. In contrast, the ones using only RAs will rank the ground-truth results in the lower-ranking positions. For instance, the fourth row shows that under the user request with only RAs: *{Shop-assembled: Yes; Classification: Desktop level},* the hypergraph-based configuration approach will return *{psb26, psb24, psb3}* as the PSBs that the user is probability interested. However, while adding US information: *'good quality'* and *'DIY for kids'* into the user request, the proposed approach then will return the PSBs *{psb3, psb16, psb0}* as the recommended results, listing the ground-truth at the top 1 position. Although the user might be unclear about the 3D printers' brand and functional specifications, they can still be recommended with proper PSBs according to their known US information.

Table 6. Examples of ranking results

User request with both RAs and USs	**Ground-truth result**	**Top 3 PSBs based on RAs and USs**	**Top 3 PSBs based on RAs**
{Classification: Industrial grade; Printing Speed: 10-150mm/s; Print Size: 305x305x300mm; User context: Good printing quality; User context: mechanically beautiful; User context: clear printing}	psb0	**psb0**	psb26
		psb26	**psb0**
		psb27	psb27

{Material: TPU, PLA, ABS; Shop-assembled: Yes; User context: need instructions; User context: easy-to-use; User context: novice user}	psb3	**psb3**	psb0
		psb25	psb1
		psb12	psb2
{Material: Photosensitive resin; Classification: Desktop level; User context: good customer service; User context: knowledgeable customer service; User context: good quality}	psb3	**psb3**	psb26
		psb0	psb24
		psb16	**psb3**
{Shop-assembled: Yes; Classification: Desktop level; User context: good quality; User context: DIY for kids}	psb3	**psb3**	psb26
		psb16	psb24
		psb0	**psb3**
{Model: Elfin; Shop-assembled: Yes; Material: Photosensitive resin; Classification: Desktop level; Time-to-market: 2019; User context: aesthetic appearance; User context: quick delivery; User context: precise printing}	psb5	**psb5**	psb26
		psb0	**psb5**
		psb9	psb27
{Shop-assembled: Yes; Material: Photosensitive resin; Classification: Desktop level; Time-to-market: 2019; User context: precise printing; User context: hot sales; User context: good customer service}	psb5	**psb5**	psb26
		psb16	psb24
		psb0	**psb5**

5.5 Comparison to the conventional PSS configuration approaches

Considering the factors related to the performance during the PSS configuration process, a qualitative comparison between the conventional PSS configuration methodologies and the proposed hypergraph-based approach was conducted using questionnaires among stakeholders. The factors selected in Table 7 are the ones that will affect the user experience throughout the PSS configuration process.

As shown in Table 7, at the beginning of the PSS configuration process, the manual configuration, the conventional feature-based PSS configuration, and ontology-based PSS configuration processes require at least basic knowledge about the PSBs; otherwise, a satisfied PSS configuration result cannot be guaranteed. However, the unbiased hypergraph-based approach with both RAs and USs relieves the reliance on domain knowledge. During the PSS configuration process, except for the quick responses to the user requests, the unbiased hypergraph-based approach also shows strength in the sensitivity to user-mentioned context details. Finally, as for the acceptance of the retrieved PSBs, all the approaches can derive proper PSBs to the users, but the proposed approach enables the more reasonable PSB ranks. Meanwhile, although manual configuration can derive precise PSB results, the configuration process duration cannot be neglected.

Table 7. Qualitative comparison in the PSS configuration process aspect

	Manual configuration by engineers/experts	**Conventional feature-based PSS configuration**	**Ontology-based PSS configuration**	**Hypergraph-based Smart PSS configuration**
Prerequisites before querying	**Poor:** Need clear RAs	**Medium:** Need domain knowledge of PSB functional specifications	**Medium:** Need domain knowledge of PSB functional specifications	**Good:** No specific requirement for the end-user
Time consumed in the configuration process	**Poor:** Take a long time to discuss with the engineers and wait for their responses	**Good:** Quick to find some necessary PSBs	**Good:** Quick to find some necessary PSBs	**Good:** Quick to find some necessary PSBs
Pertinence to the contextual	**Poor:** No contextual	**Poor:** No contextual	**Poor:** No contextual	**Good:** Sensitive to details in

information	information is involved unless initially required by the users	information is involved	information is involved	user-mentioned contexts
Acceptance of the PSB results	**Good:** Reasonable PSB are retrieved.	**Medium:** Reasonable PSBs are retrieved.	**Medium:** Reasonable PSBs are retrieved.	**Good:** Reasonable PSBs are retrieved and ranked.

6 Conclusions

In this paper, we consider a decision-making implementation framework for Smart PSS configuration under a complex situation. In Smart PSS configuration, heterogeneous entities and complicated associated relations are modelled, including RAs, USs, and PSBs, based on one-to-one, one-to-many, many-to-one, many-to-many relations. Under this situation, a hypergraph-based framework was proposed to handle the complexity in Smart PSS configuration. Moreover, facing the limitations of conventional PSS configurators and features of Smart PSS, a context-aware and self-adaptable manner for Smart PSS configuration is also achieved with the mature development of ICT techniques. The scientific contributions mainly lie in three aspects:

(1) A hypergraph-based framework was designed for the complex system containing heterogeneous entities and complicated associated relations. The framework integrates the multi-source information, including RAs, PSBs, and USs, in Smart PSS configuration with a hypergraph-based model (i.e., RA-PSB-US data model), making the configuration framework more comprehensive. The data resources of the proposed RA-PSB-US are easily accessible, thus the data model itself is generic and able to be extended. Meanwhile, by allowing users to offer their preferred usage scenarios as auxiliary information, the Smart PSS configuration also became more user-centric and user-friendly.

(2) An UHR algorithm was proposed to mitigate the bias in hypergraphs that are prone to select settled results. By normalizing the hyperedge degree, the proposed UHR can returns

personalized PSB results, thereby ensuring the validity of the practical Smart PSS configuration. Furthermore, the performance of the proposed UHR was proved based on two evaluation metrics, i.e., MAP and NDCG, compared to the other five related approaches.

(3) This study also quantitatively demonstrates the influence of US information that it can moderately improve the ranking performance. The improvement effect is more evident under more restricted configuration situations that fewer PSBs are regarded as the ones that best meet user requirements.

In summary, it is the first attempt to deploy the Smart PSS configuration task on hypergraph by clarifying the configuration process in Smart PSS, defining the RA-PSB-US data model, and offering an idea on selecting customized PSBs.

Except for the theoretical contributions of the proposed decision-making framework within a complex system containing heterogeneous entities and complicated associated relations, the hypergraph model can also fit the features of Smart PSS. In particular, by introducing US information, the advances of context-awareness were proved via the comparison between non-context-aware models (i.e., AVG, ICF, and SVD) and context-aware models (i.e., PR, HR, and UHR) quantitatively. Meanwhile, the advances were also recognized by the qualitative comparison with conventional feature-based PSS configuration and ontology-based PSS configuration. Moreover, the self-adaptability of Smart PSS was also achieved by returning personalized instead of ‘off the shelf’ PSB configuration results. It was proved via the examples in the case study that the proposed model can properly adjust the PSB result list given different user queries.

This study still has some limitations. One limitation is that a more comprehensive and larger dataset cannot be collected and tested due to restricted data access. Even so, the dataset in this study still fits for the small or medium enterprises that only have dozens of PSBs in their product-service family. Another limitation is that hypergraph-based models are not the only ones suitable for the multi-objective decision-making problem. Integrating the information of multiple graphs is also practicable to handle the heterogeneous information. In the future, we hope to explore a unified model to aggregate the heterogeneous information of multiple graphs for the context-aware Smart PSS configuration task. Furthermore, it is hoped to adopt the proposed Smart PSS configuration framework into the other Smart PSS cases with other smartness dimensions, e.g., a higher level of self-adaptability. To extend the hypergraph-based framework's capability, other intelligent modules should be integrated as a whole. For instance, to achieve higher self-adaptability of the Smart PSS configuration, the hypergraph-based configuration model should be able to evolve the hypergraph when more configuration orders and user reviews are collected.

Acknowledgement

The authors wish to acknowledge the funding support from the National Natural Science Foundation of China (No. 52005424), and Jiangsu Provincial Policy Guidance Program (Hong Kong/Macau/Taiwan Science and Technology Cooperation, BZ2020049).

References

Alegre, U., Augusto, J.C. & Clark, T. (2016). Engineering context-aware systems and applications: A survey. Journal of Systems and Software, 117, 55-83.

Bolchini, C., Curino, C.A., Quintarelli, E., Schreiber, F.A. & Tanca, L. (2007). A data-oriented survey of context models. ACM Sigmod Record, 36(4), 19-26.

Bolchini, C., Quintarelli, E. & Tanca, L. (2013). CARVE: Context-aware automatic view definition over relational databases. Information Systems, 38(1), 45-67.

Brand, M. (2003). Fast online SVD revisions for lightweight recommender systems. In: Proceedings of the 2003 SIAM International Conference on Data Mining (SDM) (pp. 37-46): Society for Industrial and Applied Mathematics.

Champiri, Z.D., Shahamiri, S.R. & Salim, S.S.B. (2015). A systematic review of scholar context-aware recommender systems. Expert Systems with Applications, 42(3), 1743-1758.

Chang, F., Zhou, G., Xiao, X., Tian, C. & Zhang, C. (2018). A function availability-based integrated product-service network model for high-end manufacturing equipment. Computers & Industrial Engineering, 126, 302-316.

Chen, D., Chu, X., Su, Y. & Chu, D. (2014). A new conceptual design approach for context-aware product service system. In: 2014 IEEE International Conference on Industrial Engineering and Engineering Management (pp. 1389-1393).

Chien, C.-F., Kerh, R., Lin, K.-Y. & Yu, A.P.-I. (2016). Data-driven innovation to capture user-experience product design: An empirical study for notebook visual aesthetics design. Computers & Industrial Engineering, 99, 162-173.

Curino, C., Quintarelli, E. & Tanca, L. (2006). Ontology-based information tailoring. In: 22nd International Conference on Data Engineering Workshops (ICDEW'06) (pp. 5-5): IEEE.

Dey, A.K. (2001). Understanding and Using Context. Personal and Ubiquitous Computing, 5(1), 4-7.

Dou, R., Huang, R., Nan, G. & Liu, J. (2020). Less diversity but higher satisfaction: An intelligent product configuration method for type-decreased mass customization. Computers & Industrial Engineering, 142, 106336.

Dou, R., Zhang, Y. & Nan, G. (2016). Customer-oriented product collaborative customization based on design iteration for tablet personal computer configuration. Computers & Industrial Engineering, 99, 474-486.

Dunke, F. & Nickel, S. (2020). A data-driven methodology for the automated configuration of online algorithms. Decision Support Systems, 137, 113343.

Guillon, D., Ayachi, R., Vareilles, É., Aldanondo, M., Villeneuve, É. & Merlo, C. (2020). ProductϒService system configuration: a generic knowledge-based model for commercial offers. International Journal of Production Research, 1-20.

Javadian Sabet, A., Rossi, M., Schreiber, F.A. & Tanca, L. (2020). Context awareness in the travel companion of the shift2rail initiative. In: 28th Italian Symposium on Advanced Database Systems, SEBD 2020 (Vol. 2646, pp. 202-209): CEUR-WS.

Kim, Y.S., Wang, E., Lee, S.W. & Cho, Y.C. (2009). A product-service system representation and its application in a concept design scenario. In: Proceedings of the 1st CIRP Industrial Product-Service Systems (IPS2) Conference: Cranfield University Press.

Lee, J. & Kao, H.-A. (2014). Dominant Innovation Design for Smart Products-Service Systems (PSS): Strategies and Case Studies. In: 2014 Annual SRII Global Conference (SRII) (pp. 305-310): IEEE.

Li, X., Chen, C.-H., Zheng, P., Jiang, Z. & Wang, L. (2021a). A context-aware diversity-oriented knowledge recommendation approach for smart engineering solution design. Knowledge-Based Systems, 215, 106739.

Li, X., Wang, Z., Chen, C.-H. & Zheng, P. (2021b). A data-driven reversible framework for achieving Sustainable Smart product-service systems. Journal of Cleaner Production, 279, 123618.

Li, X., Wang, Z., Zheng, P., Peng, T. & Chen, C.H. (2020). A Closed-Loop Context-Aware Framework for Sustainable Smart PSS Development. In: 2020 IEEE 16th International Conference on Automation Science and Engineering (CASE) (pp. 311-316).

Li, Y., Chu, X., Chu, D. & Liu, Q. (2014). An integrated module partition approach for complex products and systems based on weighted complex networks. International Journal of Production Research, 52(15), 4608-4622.

Long, H.J., Wang, L.Y., Shen, J., Wu, M.X. & Jiang, Z.B. (2013). Product service system configuration based on support vector machine considering customer perception. International Journal of Production Research, 51(18), 5450-5468.

Long, H.J., Wang, L.Y., Zhao, S.X. & Jiang, Z.B. (2016). An approach to rule extraction for product service system configuration that considers customer perception. International Journal of Production Research, 54(18), 5337-5360.

Mannweiler, C. & Aurich, J.C. (2011). Customer oriented configuration of product-service systems. In: Functional Thinking for Value Creation (pp. 81-86): Springer.

Mao, M., Lu, J., Han, J. & Zhang, G. (2019). Multiobjective e-commerce recommendations based on hypergraph ranking. Information Sciences, 471, 269-287.

Mihalcea, R. & Tarau, P. (2004). Textrank: Bringing order into text. In: (pp. 404-411).

Pan, Y., Wu, D. & Olson, D.L. (2017). Online to offline (O2O) service recommendation method based on multi-dimensional similarity measurement. Decision Support Systems, 103, 1-8.

Pradeep, P. & Krishnamoorthy, S. (2019). The MOM of context-aware systems: A survey. Computer Communications.

Ren, J., Long, J. & Xu, Z. (2019). Financial news recommendation based on graph embeddings. Decision Support Systems, 125, 113115.

Rijsdijk, S.A. & Hultink, E.J. (2009). How Today's Consumers Perceive Tomorrow's Smart Products. Journal of Product Innovation Management, 26(1), 24-42.

Rolls-Royce. Power by the hour. In: (Vol. 2021).

Rose, S., Engel, D., Cramer, N. & Cowley, W. (2010). Automatic keyword extraction from individual documents. Text mining: applications and theory, 1-20.

Roy, R., Shehab, E., Tiwari, A., Aurich, J.C., Wolf, N., Siener, M. & Schweitzer, E. (2009). Configuration of product-service systems. Journal of Manufacturing Technology Management.

Sarwar, B., Karypis, G., Konstan, J. & Riedl, J. (2001). Item-based collaborative filtering recommendation algorithms. In: Proceedings of the 10th international conference on World Wide Web (pp. 285-295).

Shen, J., Erkoyuncu, J.A., Roy, R. & Wu, B. (2017). A framework for cost evaluation in product service system configuration. International Journal of Production Research, 55(20), 6120-6144.

Shen, J., Wang, L. & Sun, Y. (2012). Configuration of product extension services in servitisation using an ontology-based approach. International Journal of Production Research, 50(22), 6469-6488.

Sheng, Z., Liu, C., Song, J. & Xie, H. (2015). Module division and configuration modeling of CNC product–service system. Proceedings of the Institution of Mechanical Engineers, Part C: Journal of Mechanical Engineering Science, 231(3), 494-506.

Song, W. & Chan, F.T.S. (2015). Multi-objective configuration optimization for product-extension service. Journal of manufacturing systems, 37, 113-125.

Tan, S., Bu, J., Chen, C., Xu, B., Wang, C. & He, X. (2011). Using rich social media information for music recommendation via hypergraph model. ACM Trans. Multimedia Comput. Commun. Appl., 7S(1), Article 22.

Tukker, A. (2004). Eight types of product–service system: eight ways to sustainability? Experiences from SusProNet. Business Strategy and the Environment, 13(4), 246-260.

Valencia Cardona, A.M., Mugge, R., Schoormans, J.P.L. & Schifferstein, H.N.J. (2014). Challenges in the design of smart product-service systems (PSSs): Experiences from practitioners. In: Proceedings of the 19th DMI: Academic Design Management Conference.: Design Management Institute.

Wang, P.P., Ming, X.G., Wu, Z.Y., Zheng, M.K. & Xu, Z.T. (2014). Research on industrial product–service configuration driven by value demands based on ontology modeling. Computers in Industry, 65(2), 247-257.

Wang, Y., Li, X. & Tsung, F. (2020). Configuration-Based Smart Customization Service: A Multitask Learning Approach. IEEE Transactions on Automation Science and Engineering, 1-10.

Wang, Z., Chen, C.-H., Zheng, P., Li, X. & Khoo, L.P. (2019a). A graph-based context-aware requirement elicitation approach in smart product-service systems. International Journal of Production Research, 1-17.

Wang, Z., Chen, C.-H., Zheng, P., Li, X. & Khoo, L.P. (2019b). A novel data-driven graph-based requirement elicitation framework in the smart product-service system context. Advanced engineering informatics, 42, 100983.

Yang, D., Li, X., Jiao, R.J. & Wang, B. (2018). Decision support to product configuration considering component replenishment uncertainty: A stochastic programming approach. Decision Support Systems, 105, 108-118.

Yu, J., Tao, D. & Wang, M. (2012). Adaptive Hypergraph Learning and its Application in Image Classification. IEEE Transactions on Image Processing, 21(7), 3262-3272.

Zhang, Z., Petrak, J. & Maynard, D. (2018). Adapted TextRank for Term Extraction: A Generic Method of Improving Automatic Term Extraction Algorithms. Procedia Computer Science, 137, 102-108.

Zhang, Z., Xu, D., Ostrosi, E. & Cheng, H. (2020). Optimization of the Product–Service System Configuration Based on a Multilayer Network. Sustainability, 12(2), 746.

Zheng, P., Lin, T.-J., Chen, C.-H. & Xu, X. (2018). A systematic design approach for service innovation of smart product-service systems. Journal of Cleaner Production, 201, 657-667.

Zheng, P., Wang, Z. & Chen, C.-H. (2019a). Industrial smart product-service systems solution design via hybrid concerns. In.

Zheng, P., Wang, Z., Chen, C.-H. & Khoo, L.P. (2019b). A survey of smart product-service systems: Key aspects, challenges and future perspectives. Advanced Engineering Informatics, 42, 100973.

Zhou, D., Huang, J. & Schölkopf, B. (2007). Learning with hypergraphs: Clustering, classification, and embedding. In: Advances in neural information processing systems (pp. 1601-1608).