

Article

A Hybrid Knowledge-Based Recommender for Product-Service Systems Mass Customization

Laila Esheiba , Amal Elgammal, Iman M. A. Helal  and Mohamed E. El-Sharkawi

Faculty of Computers and Artificial Intelligence, Cairo University, Giza 12613, Egypt;
a.elgammal@fci-cu.edu.eg (A.E.); i.helal@fci-cu.edu.eg (I.M.A.H.); m.elsharkawi@fci-cu.edu.eg (M.E.E.-S.)
* Correspondence: laila.abdelrahman@fci-cu.edu.eg



Citation: Esheiba, L.; Elgammal, A.; Helal, I.M.A.; El-Sharkawi, M.E. A Hybrid Knowledge-Based Recommender for Product-Service Systems Mass Customization. *Information* **2021**, *12*, 296. <https://doi.org/10.3390/info12080296>

Academic Editors: Marco Polignano and Giovanni Semeraro

Received: 15 June 2021
Accepted: 21 July 2021
Published: 26 July 2021

Publisher's Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.



Copyright: © 2021 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

Abstract: Manufacturers today compete to offer not only products, but products accompanied by services, which are referred to as product-service systems (PSSs). PSS mass customization is defined as the production of products and services to meet the needs of individual customers with near-mass-production efficiency. In the context of the PSS mass customization environment, customers are overwhelmed by a plethora of previously customized PSS variants. As a result, finding a PSS variant that is precisely aligned with the customer's needs is a cognitive task that customers will be unable to manage effectively. In this paper, we propose a hybrid knowledge-based recommender system that assists customers in selecting previously customized PSS variants from a wide range of available ones. The recommender system (RS) utilizes ontologies for capturing customer requirements, as well as product-service and production-related knowledge. The RS follows a hybrid recommendation approach, in which the problem of selecting previously customized PSS variants is encoded as a constraint satisfaction problem (CSP), to filter out PSS variants that do not satisfy customer needs, and then uses a weighted utility function to rank the remaining PSS variants. Finally, the RS offers a list of ranked PSS variants that can be scrutinized by the customer. In this study, the proposed recommendation approach was applied to a real-life large-scale case study in the domain of laser machines. To ensure the applicability of the proposed RS, a web-based prototype system has been developed, realizing all the modules of the proposed RS.

Keywords: constraint satisfaction problem; ontology; product-service systems; product-service system customization; recommender systems

1. Introduction

Today, manufacturers are seeking to meet their orders on demand via short-term networks, in which they are negotiating value-added processes. In addition, they try to take into account customer requirements, quality, price, sustainability, and other dimensions [1–3]. To gain a competitive advantage, manufacturers are not only offering a product, but also offering products supplemented by services, called “product-service systems” (PSSs) [4].

However, PSSs suffer from a variety of drawbacks [3]. The most significant drawback is that PSSs remain at a conceptual level and lack IT implementation. Moreover, they do not fulfill the increasing user expectations or product diversity features to enable successful customization. They are unable to capture the views of different stakeholders to adapt product design to the customer's demands in real-time. PSSs do not provide a holistic view of products and services, connecting the product structure to product quality, services, and production processes.

This creates a demand for the use of novel lifecycles, techniques, and technologies to help manufacturers link their data, processes, systems, personnel to assist customers with the assistance of product designers, and engineers in designing personalized products and services. In [3], the authors introduced a novel PSS customization lifecycle with

supporting IT tools that ensures continuous collaboration between customers and product designers and guarantees that customers' preferences are taken into consideration during the PSS customization process. The PSS customization lifecycle was established based on tried and tested knowledge-intensive structures called manufacturing blueprints. These blueprints semantically capture product-service and production-related knowledge [5,6]. Manufacturing blueprints integrate distributed manufacturing data from different locations and sources to gain full visibility and control over manufacturing data and act as a guide for the production of actionable intelligence. The PSS customization lifecycle incorporates five core processes [3], i.e., smart product ideation, PSS customization, production planning, production execution, and production monitoring.

During the PSS customization lifecycle, a massive amount of product-service-related data can be collected. However, PSSs do not support the analysis of these collected data to enhance data-driven decision making. Therefore, this massive data overload issue hinders the various stakeholders (e.g., the customer, production engineer, and shop floor operator) involved in the PSS customization lifecycle from making informed decisions.

This necessitates the use of novel techniques/approaches to assist the various involved stakeholders in making informed decisions and to accelerate the different PSS customization lifecycle processes. Accordingly, this study aims to address the following research question: "How can data analytics techniques be used to support the various stakeholders involved in making informed decisions throughout the PSS customization lifecycle?"

To provide answer to this research question, data analytics techniques can be utilized to analyze this massive amount of data collected during the PSS customization lifecycle. The analysis of these data can improve data-driven decision making and identify new opportunities. This leads to smarter business moves, more satisfied stakeholders, and more efficient operations. Prescriptive analytics is the most sophisticated data analytics branch as it answers the question of "what should we do?", which adds high value to organizations. It includes optimization and simulation techniques to provide advice, explore several possible actions, and suggest a course of action, through the use of statistics and data mining techniques.

Recommender systems (RSs) belong to the class of prescriptive analytics, which represent software tools that offer customers with useful suggestions, taking into account their preferences/requirements [7]. There are several types of recommender systems. Some examples of these types are: content-based RSs [8], collaborative filtering RSs [7], hybrid RSs [8], knowledge-based RSs [9], knowledge graph-based RSs [10], and cognitive-based RSs [11]. Recommender systems have gained a prominent role due to their applicability in a variety of domains (e.g., tourism, e-commerce, e-learning, health, etc.) and the large number of applications that offer personalized services [7,12,13].

We envision that RSs could play a pivotal role throughout the different processes of the smart PSS customization lifecycle [3], that is, (i) assisting various involved stakeholders with different views throughout the smart manufacturing process in making informed decisions, and (ii) enabling the re-usability of previous successful customized PSS variants. Accordingly, in [14], we have analyzed and developed a recommendation framework that supports the different processes of the PSS customization lifecycle introduced in [3]. In this framework, a set of recommendation capabilities are identified for each process, while accommodating different stakeholders' perspectives.

The work presented in this paper focuses on the realization of the recommendation facility identified for the "smart product ideation" process of the PSS customization lifecycle [3], which we identified in [14] as "recommending previously customized product-service systems (PSSs)". During the smart product ideation process, customers may start their customization process by selecting a PSS variant from the wide range of available previously customized PSS variants maintained in the manufacturing blueprints knowledge base. Consequently, finding a PSS variant that is precisely aligned to the customers' requirements is a cognitive task that the customers are unable to manage easily.

Despite the fact that personalized recommendations have received a lot of attention in a variety of domains, such as e-commerce, tourism, telecommunications, and financial services, few previous research efforts have been directed toward PSS customization recommendations. Moreover, when generating PSS recommendations, these studies only take into account the features of the services that accompany the product. Accordingly, we propose a hybrid knowledge-based recommender for PSS customization recommendations. Our proposed approach distinguishes itself from other previous studies reported in the literature due to its capability to generate PSS recommendations while taking into account customers' requirements (e.g., functional, structural, environmental, cost, and quality) for the product and its associated services.

The backbone of the proposed recommendation approach is a knowledge base that utilizes a set of integrated ontologies called manufacturing blueprints [6], which capture rich product-service and production-related knowledge. Manufacturing blueprints turn conventional products into smart self-describing products by storing, linking, combining, and analyzing the raw data collected by the product throughout its lifecycle. In this manner, smart actionable data is created, from which knowledge can be generated and production processes can be triggered.

The proposed recommender system is a hybrid ontology-based system in the sense that it integrates and combines two techniques: (i) *constraint modeling*, where the problem of selecting previously customized PSS variants is formulated as a constraint satisfaction problem (CSP) [15], to deduce a subset of potential customized PSS variants from the wide range of available ones maintained in the manufacturing blueprints KB. (ii) A *weighted utility function* is used to rank the remaining PSS variants based on their utility to the customer. The utility, efficacy, and applicability of the proposed approach is demonstrated herein through its implementation in a real-life large-scale case study from the H2020 ICP4Life project ICP4Life: <http://www.icp4life.eu/> (accessed on 22 July 2021).

The main contributions of this paper are:

- We propose a hybrid knowledge-based recommender for PSS customization recommendations.
- The manufacturing blueprints models proposed in [5,6] are extended to support the PSS customization recommendation process.
- The utility, efficacy, and applicability of the proposed approach is carried out through its implementation in a real-life case study in the domain of laser machines.
- To ensure the applicability of the proposed recommendation approach, a web-based prototype system has been developed, realizing all the modules of the proposed RS.

The rest of this paper is organized as follows. Related work is analyzed in Section 2. In Section 3, the case study is demonstrated, whereas in Section 4, manufacturing blueprints models are discussed with extensions to support the recommendation process. Section 5 presents the hybrid knowledge-based recommendation approach, as well as its implementation and evaluation details. Finally, conclusions and future work are highlighted in Section 6.

2. Related Work

The related works focus on two directions: (i) the role of ontologies in manufacturing, and (ii) recommendation approaches in manufacturing.

2.1. Using Ontologies in Manufacturing

Ontologies are formal explicit representations of the concepts (classes) and the relationships between domain concepts [16,17]. For the manufacturing domain, ontologies are used to share knowledge between different manufacturing systems, stakeholders, and applications.

The authors in [18] proposed the Product ONTOlogy (PRONTO) approach for modelling products for the purpose of sharing product information across organizations. This method addresses concepts primarily related to the product structure, but it has no refer-

ence to any existing standards developed for the modeling of product structure, processes, and features. To overcome the lack of modeling standards in [18], the authors in [19] proposed a product ontological model based on ISO and IEC standards.

Influential work related to the conceptualization of product-centered services is reported in [20,21]. A service architecture was presented in [20], introducing an ontology model that focuses on the adaptation of services to the PSS context, but the product perspective and product-service interface are not addressed. Based on findings from industrial cases, the authors in [21] developed a knowledge structure, as well as an ontology, focusing on product-related service and service concepts, without any specific discussion about the product itself.

Some authors have utilized ontologies for modelling PSS concepts. In [22], the authors conducted extensive interviews with experts to elicit PSS root concepts in order to propose a suitable PSS ontology. The proposed ontology focuses on the top-level concepts of the PSS from a design perspective. In [23], the authors proposed a PSS ontology to manage the exchange of information among stakeholders throughout the different phases of the PSS lifecycle.

Another stream of research has utilized ontologies to represent product and/or service configuration knowledge for the purpose of configuration. In [24], the authors proposed an ontology-based configuration system for configuring product extension services (PESs) in servitization. They proposed three meta-ontologies, namely, product, service, and customer ontologies. In [25], the authors presented an ontology-based approach for the modeling of product configuration knowledge. Concepts within the product configuration domain, such as product, component, part, attribute, etc., were only considered; moreover, classes related to the configuration of product-related services were not considered.

From a methodological perspective, the authors in [3] proposed a novel product-service customization lifecycle that included technological solutions aiming to enable PSS customization. The PSS customization lifecycle relies on the tried and tested manufacturing blueprints proposed in [5,6], which provide a basis for actionable PSS and production intelligence.

A summary of the related work and ontological models used in manufacturing is provided in Table 1. Based on this summary, it was found that the proposed models were utilized for modeling product and/or service knowledge for the purposes of design, planning, or configuration. The issue of extending those models for other applications (PSS customization recommendations) addressed by this paper is an open research topic.

Table 1. Related work and ontological models used in manufacturing.

Paper	Modelling Scope	Application Domain	Evaluation Mechanism	Limitations
[3]	Product, service, partner, production-process, quality assurance	Customization of product-service systems (PSS) in the laser cutting machines domain	Laser cutting machines case study	No customer profile modeling
[18]	Product	Material requirements planning	Food industry case study	Lack of modeling standards
[19]	Product, process, resources	Solving interoperability problems between different enterprises and applications	Simulation of distributed activities for manufacturing simple product prototypes	Consideration of only two modeling standards (ICE, and ISO)
[20]	Service	Service planning	No evaluation provided	Product perspective and product-service interface are not considered

Table 1. Cont.

Paper	Modelling Scope	Application Domain	Evaluation Mechanism	Limitations
[21]	Service	Product design	Semi-conductor pump and lubricant-free small vacuum pump designing	Product-related service and service concepts are considered without any specific discussion about the product itself
[22]	Product, service, stakeholder, PSS design, PSS outcome	PSS design	Extensive interviews with domain experts	Focus on the top-level concepts of the PSS from a design perspective
[23]	Product portfolio, service portfolio	Enhancing information interoperability among stakeholders throughout the different phases of the PSS lifecycle	Bike rental case study	No evaluation was conducted with industrial experts
[24]	Product, service, customer	Configuring product extension services (PES) in servitization	Example of configurable product-extension services called “building solution”	Only feasible configuration solutions are considered, whereas optimal ones have not yet been considered
[25]	Product	Product configuration	Ranger drilling machine case study	Only feasible configuration solutions are considered, whereas optimal ones have not yet been considered

2.2. Recommendation Approaches in Manufacturing

Recommender systems (RSs) are software tools and techniques, utilized to assist customers in making the right decisions [7]. RSs have been successfully applied in various domains, such as tourism, health, e-commerce, e-learning, etc. There are several types of recommendation techniques; the foremost common techniques are:

- The collaborative filtering technique: predicts what would be of interest to a person based on the taste of many other users [7];
- The content-based technique: The content-based technique relies on the features of products and the user preferences. This technique recommends products with similar features to those enjoyed by customers in the past [8];
- The knowledge-based technique: generates suggestions based on the domain knowledge and user explicit requirements. This approach does not consider the behavior of other users [9]. Knowledge-based techniques are divided into the constraint-based technique [26] and the case-based reasoning (CBR) technique [27]. The constraint-based technique exploits a predefined recommender knowledge base, which contains explicit constraints about how to relate customer requirements with product features. The CBR technique generates recommendations based on similarity metrics;
- Hybrid techniques: based on combining two or more recommendation techniques into one hybrid technique to gain better performance [7].

Content-based and collaborative filtering approaches are suitable for domains where a massive amount of data and individual interactions exist. In the context of the PSS customization domain, these approaches are not the leading choice, as there is insufficient data available on actual user selections for preferred customized PSS variants.

Knowledge-based techniques help to tackle the cold start problem [28], which deals with the absence of data about past user choices. This problem is tackled through combining customers’ explicit requirements, as stated during the recommendation session, and supported in our solution by the manufacturing blueprints and the related knowledge base.

Therefore, our proposed approach (cf. Section 5 exploits knowledge-based techniques and more specifically the constraint-based technique for recommending previously customized PSSs in the mass customization domain.

Recommender systems have been successfully applied in the manufacturing domain. In the domain of additive manufacturing (AM), the authors in [29] proposed a hybrid machine learning approach for recommending additive manufacturing design features for target components in the conceptual design phase. The proposed approach combines both clustering and support vector machine (SVM) algorithms for the generation of recommendations. A case study of designing R/C racing car components was used to validate the proposed recommendation approach.

Another stream of research has utilized RSs in the domain of cloud manufacturing (CMfg) to assist customers in identifying the best manufacturing services (i.e., resources and capabilities) to accomplish the required manufacturing task [30–33].

In [30], a hybrid approach that integrates social network and collaborative filtering (CF) techniques was proposed to predict the missing quality of service (QoS) values of manufacturing services. Based on this prediction, top-k optimal manufacturing services with higher QoS values are recommended to service consumers. Similarly, in [31], a novel clustering-based and trust-aware RS was developed for reliable cloud manufacturing service recommendations. The purpose of this system is to predict the QoS values of the cloud manufacturing services and allow users to retrieve the most relevant services.

In [32], the authors proposed a novel manufacturing service recommendation algorithm based on a time-aware targeted reconstructing service descriptions (T-TRSD) model. The purpose of T-TRSD is to reconstruct the single manufacturing service descriptions for specific requirements, taking into account the evolving characteristics and service composition descriptions of cloud manufacturing services. The recommendation of manufacturing services is eventually carried out by mining the valuable information contained in the reconstructed manufacturing service descriptions.

In [33], the authors proposed an intelligent RS for the recommendation of cloud manufacturing services. The proposed RS adopts a deep neural network (DNN) paradigm, which allows the automatic learning of an optimal manufacturing services list based on customers' past experiences and new choices.

Although personalized recommendations have received a great deal of attention in a variety of domains, such as telecommunications [34], tourism [35,36], financial services [37,38], and e-commerce [39–41], few applications of RSs for personalized PSS recommendations have been reported in the very recent literature. In [42,43], the authors proposed a multi-criteria recommendation method based on a rough collaborative filtering (CF) approach to provide customers with customized PSS solutions. However, the recommendation of these customized PSS solutions depends on product-service features only, such as service response time, service cost, service reliability, etc. The authors do not take into considerations the structural and quality characteristics of the product as a PSS component.

Table 2 summarizes the related work pertaining to recommendation approaches in manufacturing. Based on this summary, it was found that: (i) the majority of previous studies concentrate on recommending the best manufacturing services for accomplishing a manufacturing task in the domain of cloud manufacturing, (ii) few previous researches are targeted for PSS customization recommendations; moreover, the recommendation of PSS solutions in these studies has been carried out based on product-service features only.

Table 2. Related work pertaining to recommendation approaches in manufacturing.

Paper	Applied Techniques	Application Domain	Recommendation Capabilities	Evaluation Mechanism
[29]	Clustering algorithm, support vector machine (SVM) classification algorithm	Additive manufacturing	Additive manufacturing design features	R/C car racing components case study
[30]	Social network, collaborative filtering (CF)	Cloud manufacturing	Cloud manufacturing services	Experimental evaluation
[31]	Clustering algorithm, CF approach	Cloud manufacturing	Cloud manufacturing services	Experimental evaluation
[32]	Time-aware targeted reconstructing service descriptions (T-TRSD) model	Cloud manufacturing	Cloud manufacturing services	Experimental evaluation on a real-world data set
[33]	Deep neural network (DNN) approach	Cloud manufacturing	Cloud manufacturing services	Simulated case study
[42,43]	Decision-making and trial evaluation laboratory (DMATEL) method, collaborative filtering (CF)	PSS customization	Customized PSS solutions	Elevator case study

Unlike previously proposed approaches for PSS customization recommendations in the literature, which consider only product-service features for recommending PSS solutions, our proposed approach in this article takes into consideration customer requirements (functional, structural, environmental, cost, and quality requirements) for the product and its associated services when calculating PSS recommendations.

3. Case Study

In this section, we present a case study that is conducted in the context of the EU H2020 ICP4Life [3]. This case study was carried out for a turbine engine manufacturer (customer) who was interested in a multi-axis laser machine. In a previous work [3], the goals were concerned with co-designing and identifying the laser machine components based on the customer requirements using the novel product-oriented configuration language (PoCL) [1]. PoCL is a model-based graphical user-friendly domain-specific language (DSL) which aims to ease the task of collaborative product design by using the same jargon that is familiar to customers and other stakeholders. By using this user-friendly language, the customer collaborates with the product designer to elicit and validate the desired characteristics of the product and its associated services. The output of this customization process is a set of customization requirements and design parameters concerning the multi-axis laser machine and its associated services, which are validated, transformed, and eventually stored as a new customized PSS in the blueprint knowledge base for further re-usability, using the previously developed tool-suite [1].

From a recommendation perspective, which is the primary goal of our paper, previously customized PSS variants maintained in the blueprint knowledge base could be re-used as a starting point for a new customization request. In this case, the customer uses a web application to specify the laser machine requirements, parts, and preferences. Furthermore, the customer determines her business environment properties (i.e., business type, environment temperature).

For example, the customer may specify that her business environment temperature is high, that the laser machine features should include a CO₂ laser generator, with a high

power and speed, and that the work piece should be fixed. The customer may also request to include services such as maintenance, repair, delivery, and installation, etc. In addition, the customer may indicate her budget constraints with respect to the product and its associated services.

Moreover, the customer may specify her preferences in a set of laser machine utility dimensions, such as reliability, performance, etc. She can specify her preferences in terms of the importance (weight) of each utility dimension. For example, the customer may specify that she is highly interested in *reliability* with an importance weight of 0.5.

By utilizing the proposed recommendation approach, a set of previously customized PSS variants are recommended to the customer based on her requirements and preferences. The incorporation of this recommendation approach assists customers in accurately finding the required PSS variant with lower search costs.

4. Manufacturing Blueprints

Manufacturing Blueprints [5,6] are knowledge templates that capture rich product-service and production-related knowledge. This enables automated reasoning and inference to validate the consistency of the customized PSS, verify customer constraints and preferences, and generate a preliminary production plan. Manufacturing blueprints, which rely on model-based design techniques to manage and inter-link product data, information (both its content and context), product portfolios and product families, manufacturing assets (personnel, plant machinery and facilities, production line equipment), production processing requirements, and workflows. Manufacturing blueprints help meet the requirements (functional, performance, quality, cost, physical factors, interoperability, time, etc.) of an entire manufacturing network. This information can be collated and put within a broader operational context, providing the basis for production actionable “intelligence” and a move toward more fact-based manufacturing decisions.

Manufacturing knowledge is encapsulated in the five inter-connected knowledge-based templates:

- Supplier blueprint: containing knowledge about the supplier’s firm, its capabilities, and details.
- Product blueprint: includes information about the product, its components, quality attributes, and product families.
- Product-service blueprint: defines the characteristics of all services that are coupled with the physical product, such as service quality attributes.
- Production process blueprint: describes the workflow of the process, the activities involved, and the resources required for actual production.
- Quality assurance blueprint: defines the KPIs needed to monitor the production processes and to solve production problems.

To meet the recommendation of PSS customization during the smart product ideation process, we have extended the manufacturing blueprints models, as will be discussed further below.

The product blueprint proposed in [6] is extended to integrate appropriate classes that are needed to meet customers’ requirements and facilitate the recommendation of PSS customization. Examples of the classes involved in this blueprint are: (i) product: a general class used to describe a product made up of parts and sub-assemblies; (ii) product component: this class describes the components that make up the product; (iii) operational environment: is a new added class to the product blueprint that describes the properties of the environment in which a product can operate (e.g., environment temperature, environment humidity). With the addition of this class, we are now able to recommend products that meet the customers’ environmental constraints.

Moreover, we need to model the customer profile before recommending a previously customized PSS, so a new knowledge structure/template is added, which is called the customer blueprint. The customer profile consists of two parts: (i) the customer’s basic information (e.g., location, business environment, business type, business size, etc.)

and (ii) the customer's requirements. Figure 1 illustrates the main classes involved in the customer blueprint and the relationships between them. The customer has a set of requirements specifying her request. The requirements class is classified into four types of requirements:

- The ComponentRequirements class: represents the components that the customer may request. This class is further sub-divided into two classes: (i) the *ServiceComponentRequirements* class, representing the service components that the customer may request (e.g., maintenance); and (ii) the *ProductComponentRequirements* class, representing the product's components that may be required by the customer (e.g., laser head).
- The AttributeRequirements class: represents the features of the product/service/component that may be required by the customer. This class is sub-divided into four classes: (i) the *ProductAttributeRequirements* class, describing the attributes associated with the product (e.g., product reliability) that may be required by the customer; (ii) the *ProductComponentAttributeRequirements* class, representing the attributes of the product's components that may be required by the customer (e.g., component speed, power); (iii) the *ServiceAttributeRequirements* class, describing the attributes related to the services that may be required by the customer (e.g., service response time, service performance); (iv) the *ProductFunctionAttributeRequirements* class, describing the attributes related to a certain product function. In the context of the laser machine domain, the product may have a certain function, such as drilling, cutting, or welding. These functions have a set of attributes, such as speed and productivity, which are described by this class.
- The FunctionalRequirements class: describes the function required by the customer from the product or its associated services. This class is sub-divided into two sub-classes: (i) the *ProductFunctionRequirements* class, describing the product function required by the customer; (ii) the *ServiceFunctionRequirements* class, describing the service function required by the customer (e.g., maintaining the laser head).
- The CostRequirements class: this class specifies the customer's acceptable cost for the product and its associated services.

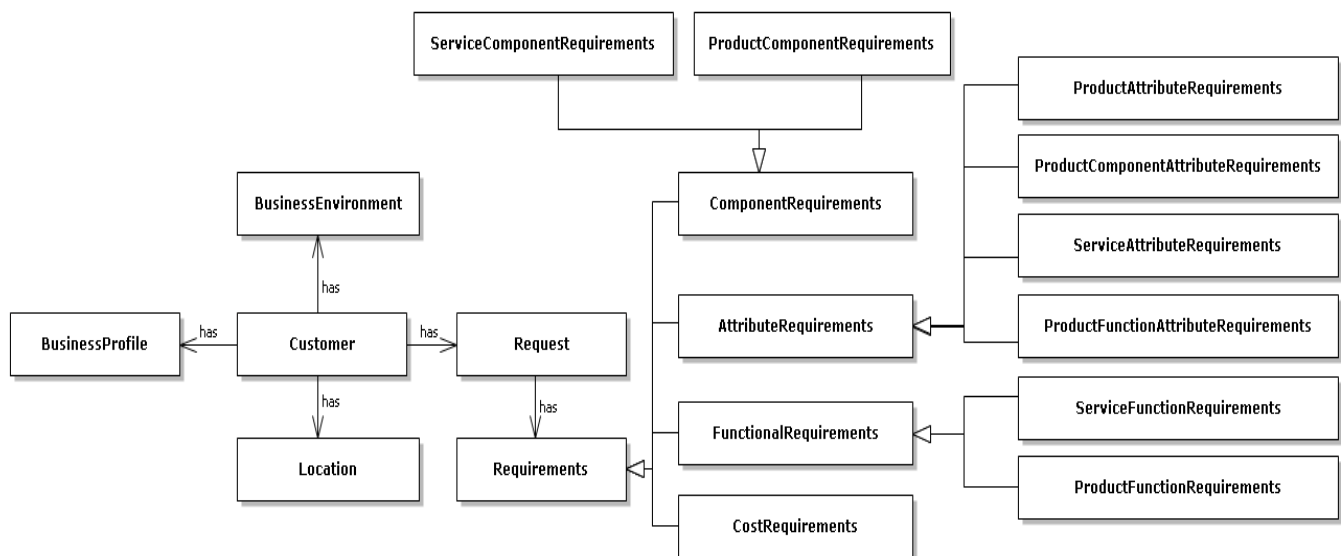


Figure 1. The customer blueprint.

In addition to the previously described classes, some other classes are used to demonstrate the basic information about the customer, such as:

- Customer class: represents the customer who requests the product and its associated services.

- Business profile class: contains information about the customer's business profile (e.g., business size, business type, etc.).
- Business environment class: contains information about the customer's business environment properties (e.g., temperature, humidity).
- Location class: represents information about the customer's location.

5. Proposed Recommendation Approach

In this section, we focus on the realization of the recommendation facility identified for the “smart product ideation” process as part of the PSS customization lifecycle [3], which we referred to in [14] as “recommending previously customized product-service systems (PSSs)”. In the context of PSS customization, customers are overwhelmed by a multitude of previously customized PSS variants with varying characteristics. Consequently, finding a PSS variant that is aligned with customer requirements quickly and accurately is a cognitive task that customers will not be able to manage easily. Accordingly, we propose a hybrid knowledge-based recommender system for recommending previously customized PSS variants. This recommender is hybrid in nature as it encodes the problem of selecting previously customized PSSs as a constraint satisfaction problem (CSP) [15] and uses explicit feedback from customers to generate a ranked list of customized PSS variants using a utility function.

Our proposed approach tackles the cold start problem [28] as it can generate recommendations for new users for whom the system has no information about their last choices/preferences. This is achieved through combining users' explicit requirements during the recommendation session. However, our approach suffers from the knowledge acquisition bottleneck, which means that it depends on knowledge engineers, who are required to transform the domain knowledge provided by domain experts (items' properties and the corresponding constraints) into formal representations. Knowledge engineers may not always be available, and when many constraints must be defined, the knowledge acquisition process can become more complicated. To address the aforementioned limitations in the future, we plan to rely on human computation concepts to integrate domain experts more deeply into the development and maintenance of knowledge bases. This can be done by replacing the complex tasks of knowledge engineers with simple micro-tasks [44,45] that can be performed easily, even by domain experts without technical expertise.

Figure 2 provides an overview of the proposed recommendation approach. Assume that we have a great deal of previously customized PSS variants to be offered. Our knowledge-based approach exploits the recommender knowledge base that contains a set of integrated manufacturing blueprints (as discussed in Section 4) and a set of explicit domain constraints. These constraints are defined by knowledge engineers who have knowledge about the field and are used to relate customer requirements (customer variables V_C) with PSS variables (product variables (V_{PROD}) and the associated service variables (V_{SER})). Meanwhile, the customer requirements are acquired during the recommendation session and are maintained in the knowledge base by using the customer blueprint as discussed in Section 4.

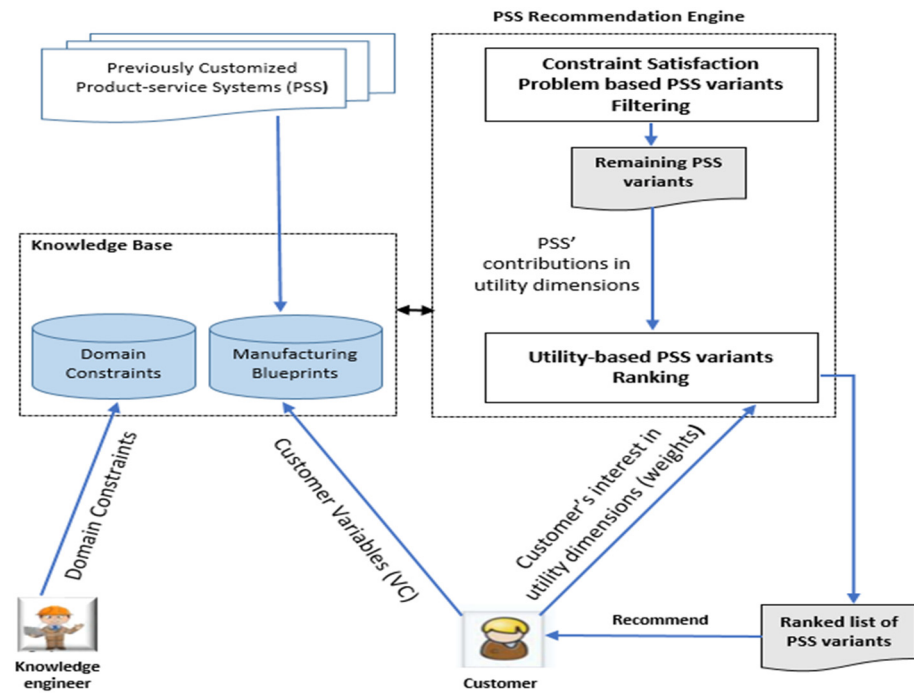


Figure 2. An overview of the proposed recommendation approach.

The proposed recommendation approach integrates two techniques: (i) the CSP is integrated into the recommender engine to filter out PSSs that do not satisfy the constraints (i.e., the customer requirements). (ii) A weighted utility function is used to rank the remaining PSS variants and these are finally presented to the customer. The previous two techniques are discussed in detail in the following two sub-sections.

5.1. Constraint Satisfaction Problem (CSP)-Based PSS Variant Filtering

The knowledge base of any constraint-based recommender system can be defined as a set of variables and a set of constraints [26]. These variables and constraints constitute the main elements of a constraint satisfaction problem (CSP) [15]. CSP can be defined as a triple (V, D, C) where V is a set of finite domain variables $\{v_1, v_2, v_3 \dots v_n\}$, D represents a set of domains from which variables in V take values $\{dom(v_1), dom(v_2), \dots, dom(v_n)\}$, and C represents a set of constraints that define restrictions on the possible combinations of variable values $\{c_1, c_2, \dots, c_m\}$ [46]. The solution of a CSP can be defined as a concrete instantiation of the variables in V such that all the specified constraints in C are fulfilled.

By following the CSP formalism, the task of recommending previously customized PSS variants is defined as a CSP, consisting of three sets of variables (V_C, V_{PROD}, V_{SER}) and four sets of constraints (C_R, C_O, C_{PSS}, C_F). These variables and constraints are defined as follows:

- *Customer variables V_C* : in the context of the laser machine domain discussed in Section 3, customer variables are subdivided into two sets of variables (V_I, V_P); where V_I is a set of variables used in capturing customer requirements and V_P is a set of variables used in capturing customers' business environmental properties. Examples of (V_I) and (V_P) are shown in Tables 3 and 4, respectively.
- *Product Variables, V_{PROD}* , describe the attributes of the offered products. V_{PROD} are divided into five sets of variables ($V_S, V_F, V_{NF}, V_E, V_M$); where V_S describes the structural properties of products, V_F describes the functional properties of products, V_{NF} refers to products' non-functional properties, V_E describes products' environmental characteristics, and V_M describes the economical properties of products, such as price. Examples of product variables are shown in Table 5.

- *Product-Service Variables, V_{SER}* , describes the attributes of the product's associated services. These variables are divided into structural (V_{SS}), functional (V_{SF}), non-functional (V_{NF}), and economic variables (V_{SE}). Examples of product-service variables are shown in Table 6.
- *Filtering constraints C_F* : a set of constraints used to define the relationship between customer and PSS variables. The values of PSS variables are constrained by customer variables. A set of filtering constraints are declared, using an object annotation language (cf. Table 7). This declaration is carried out to differentiate between customer and PSS variables using (user model) for customer variables and (PC product) for previously customized PSS variables.
- *C_O Constraints*: these constraints restrict the possible requirements of the customer. These constraints are divided into require constraints (C_{req}) and (in) compatibility constraints (C_{comp}), as shown in Table 8.
- *Previously Customized PSS, (C_{PSS})*, refers to the allowed instantiations of both product variables (V_{PROD}) and product-associated service variables (V_{SER}), which define the set of available previously customized PSSs (cf. Table 9).
- *Customer Requirements, (C_R)*, constraints are a set of customer requirements. These requirements are captured through a user interface.

Table 3. Examples of customer requirement variables.

Customer Interests (V_I)	Domain Values of V_I
Laser generator	CO2 laser/YAG laser
Laser power	low/medium/high
Focus lens	Text indicating the type of the focus lens
Product service	Maintenance/delivery and installation/recycling
Max. product price	Integer indicating the price of the product
Max. service cost	Integer indicating the cost of the service

Table 4. Examples of customer's business environment variables.

Customer Properties (V_P)	Domain Values of V_P
Business type	Drilling/welding/cutting/engraving
Customer's environmental temperature	Low/medium/high

Table 5. Examples of product variables.

Product Variables (V_{PROD})	Values of V_{PROD}
Structural Properties (V_S)	
ID	Integer (1–20)
Laser generator	Co2 laser/YAG laser
Functional Properties (V_F)	
Product function	String indicating the product's function (cutting/drilling/welding/engraving)
Non-Functional Properties (V_{NF})	
Product reliability	Float indicating the product's reliability
Environmental Properties (V_E)	
Product environment temperature	Integer indicating the operational temperature of the product's environment
Economic properties (V_M)	
Product price	Integer (e.g., 2,000,000)

Table 6. Examples of product-service variables.

Product-Service Variables (V_{SER})	Values of V_{SER}
Structural Properties (V_{SS})	
ID	Integer
Product service	String indicating the attached product's service (maintenance, spare part replacement, recycling)
Functional Properties (V_{SF})	
Service function	String indicating the product's service function (i.e., maintaining laser head)
Non-Functional Properties (V_{NF})	
Service response rate	Float indicating the response rate of the service
Service performance	Float indicating the performance rate of the service
Economic properties (V_{SE})	
Service cost	Integer indicating the cost of the service

Table 7. Examples of domain filtering constraints.

Id	Constraint
C_{F1}	If user model. Business type = 'Cutting Wood' then PC product. Capability = 'Cutting Wood'
C_{F2}	If user model. Laser generator = 'CO2 laser' then PC product. Product component. Name = 'CO2 laser'
C_{F3}	If user model. Laser power = 'High' then PC product. Laser power ≥ 3000
C_{F4}	If user model. Laser speed = 'Low' then PC product. laser speed < 100
C_{F5}	If user model. Service = 'Maintenance' then PC product. Service = 'Maintenance'
C_{F6}	If user model. Environment. Temperature = 'low' then PC product. Operational environment. Temperature < 40
C_{F7}	If user model. Max. Product price = (specific price) then PC product. Price $\leq$ (specific price)
C_{F8}	If user model. Max. Product service cost = (specific price) then PC product service. Cost $\leq$ specific price

Examples of domain constraints are defined in Table 7. For example, C_{F1} means that when the business type of the customer is "Cutting Wood", then the laser machine function should support this business type.

Table 8. Examples of require and compatibility constraints.

ID	Require Constraints
C_{req1}	YAG laser generator requires YAG beam delivery system
C_{req2}	YAG laser generator requires maintenance
C_{req3}	Fixed work piece requires fixed workpiece positioning system
C_{req4}	High laser speed requires high laser power
C_{req5}	CO2 laser generator requires CO2 beam delivery system
Compatibility Constraints	
C_{comp1}	YAG laser generator is incompatible with motor 1

Based on the previously provided definition of the recommendation task as a CSP, we can generate recommendations for the customer, given their requirements. The solution of this task is defined as a concrete instantiation of the variables (V_C , V_{PROD} , V_{SER}) such that this instantiation does not violate any of the constraints in (C_R , C_O , C_{PSS} , C_F). The CSP solver can result in more than one solution or none.

5.2. Utility-Based PSS Ranking

As mentioned before, the solutions for a CSP may be more than one solution or none. In cases where there are more than one solution, our recommendation approach utilizes a weighted utility function that is based on the multi-attribute utility theory (MAUT) approach [47]. This function is used to rank the retrieved previously customized PSS variants that satisfy the constraints (i.e., the output of the CSP-based PSS filtering process) based on their utility to the customer.

In order to calculate the utility of these retrieved PSS variants, it is crucial to have knowledge about: (i) PSS variants' contributions in utility dimensions/attributes. PSS quality attributes (e.g., *product performance*, *product reliability*, *service response time*) that are captured using the product and product-service blueprints are utilized for this purpose. (ii) The customer's interest in terms of the importance/weight of each utility attribute. Based on this information, we adopt the following weighted utility function [47] (cf. Equation (1)) for calculating the utility of PSS variants.

$$utility(PSS) = \sum_{i=1}^n w_i s_i(PSS) \quad (1)$$

where n represents the number of utility attributes, $utility(PSS)$, represents the utility of a customized PSS variant, w_i represents the customer's interest in terms of weight in an attribute i , and $s_i(PSS)$ is the contribution of the PSS variant to the attribute i . The values of w_i are acquired directly from customers during the recommendation session.

In cases where there is no matching PSS variant, we propose an algorithm that handles this case (Algorithm 1). This algorithm is based on dividing the customer requirements' constraints into weak constraints (C_w) and hard constraints (C_h). Hard constraints represent the customer's business environment (i.e., business type, business environment temperature); these constraints cannot be changed when no solution is found. Weak constraints act as the customer requirements from PSS components and their features (e.g., laser generator type, laser speed). Algorithm 1 shows how to deal with the *no solution found case* that is returned by the CSP solver.

Algorithm 1 Handling cases in which no solution is found.

Input: A set of customer requirements $C_{reqs} = \{req_1, req_2 \dots req_n\} = \{C_h \cup C_w\}$, customer interest in each requirement (weights), customer interest in each utility attribute (weights)

Output: Top-k PSS variants

```

1, Begin:
2,   SolutionList  $SL$  = GetSolutions ( $C_{reqs}.C_h$ ); /* Using CSP solver */
3,   if ( $SL$  is not empty) then
4,     return  $SL$ ;
5,   else:
6,     return 'no solution found';
7,     for each PSS variant ( $P$ ) in  $SL$  do:
8,       Get PSS feature vector ( $P$  features);
9,       Calculate Similarity ( $C_{reqs}$ ,  $P$  features) using Equation (2);
10,    end for;
11,    Sort PSS variants in  $SL$  w.r.t Similarity descendingly;
12,    Get top-N similar PSS variants to customer's requirements
13,    for each PSS variant in top-N similar list do:
14,      Calculate the utility of PSS variant using Equation (1);
15,    end for;
16,    Sort PSS variants w.r.t utility descendingly;
17,    return Top-K PSS variants based on the utility;
18, End;

```

The input to the algorithm is a set of customer requirements (C_{reqs}). First, the algorithm starts by searching for PSS variants that satisfy the customer's hard requirements (C_h) (line 2). If there exist PSS variants that satisfy the hard requirements, then the similarity between all features of these PSS variants and customer's requirements is calculated using a weighted similarity function (cf. Equation (2)). This similarity function is based on the weighted Euclidean distance [48], to measure the similarity between the customer's requirements vector and the PSS variant vector after normalizing all vectors.

$$similarity(C, P) = \sqrt{\sum_{i=1}^f w_i (C_i - P_i)^2} \quad (2)$$

where C and P are the two input vectors (customer's requirements feature vector and PSS variant feature vector), w_i defines the weight of the feature i , and f defines the number of features. Each feature is given a different weight in the customer-PSS variant similarity calculation, and this is determined by each customer's willingness to include this feature in her PSS variant. After calculating the similarity between all PSS variants and the customer's requirements, PSS variants are sorted in a descending order based on the similarity. Then, the top- N similar PSS variants to the customer's requirements are retrieved (line 7 to line 12). This is followed by the calculation of the utility of each PSS variant in the top- N similar list, using Equation (1) (line 13 to line 15). Finally, PSS variants are sorted based on their utility to the customer and top- K PSS variants are returned if a solution is found.

Reverting to the case study in Section 3, assume that there exists a set of previously customized PSS variants maintained in the manufacturing blueprint KB. Examples of previously customized PSS variants are shown in Table 9. Moreover, assume that the turbine engine manufacturer (customer) indicates her requirements as shown in Table 10.

Table 9. Examples of previously customized PSSs (C_{PSS}).

Id	Laser Generator	Workpiece	Focus Lens	Laser Power	Laser Speed	Service	Service Price	Product Environmental Temp.	Product-Price
P ₁	CO2	controllable	FL 7.5	3000	200	Maintenance	10,000	40	150,000
P ₂	YAG	fixed	FL 7	3000	200	Maintenance	9000	50	100,000
P ₃	CO2	fixed	FL 7.5	4000	250	Maintenance	10,000	60	100,000
P ₄	CO2	fixed	FL 7.5	3000	200	Maintenance	10,000	65	100,000
P ₅	CO2	fixed	FL 6	2000	100	Delivery and Installation	9000	40	75,000

Table 10. Examples of customer's requirements.

Laser Generator	Workpiece	Focus Lens	Laser Power	Laser Speed	Service	Service Max. Price	Customer Environment Temperature	Product Max. Price
CO2 laser	Fixed	FL 7.5	High	High	Maintenance	10,000	High	100,000

The customer's requirements, along with the domain filtering constraints (cf. Table 7) and previously customized PSSs, are used by the CSP solver to identify PSS variants that satisfy the customer's requirements. Based on our example scenario, the CSP solver returns two solutions (P₃ and P₄), (cf. Table 9), that satisfy the constraints and are aligned to the customer's requirements.

Therefore, we utilize a utility function (cf. Equation (1)) to rank the PSS variants retrieved from the CSP solver based on their utility to the customer. Assume that the contribution of P_3 and P_4 to domain-specific utility dimensions/attributes is as shown in Table 11.

Table 11. PSS variants' contributions to domain utility dimensions.

Id	Reliability	Performance
P_3	0.6	0.8
P_4	0.8	0.8

In addition, assume that the interest (importance weight) of the customer in each dimension is as follows: *Reliability weight* = 0.5, *Performance weight* = 0.5; the utility of PSS variants P_3 and P_4 is calculated using Equation (1). Then, these variants are ranked and presented in a tabular form in descending order based on their utility to the customer, as shown in Table 12.

Table 12. Ranked list of PSS variants based on their utility to the customer.

Id	Utility to the Customer
P_4	0.8
P_3	0.7

5.3. Implementation

In order to ensure the applicability of the proposed approach, a web-based prototype system has been developed as a proof-of-concept. The prototype implements all the modules introduced in the proposed recommender system architecture (cf. Figure 3).

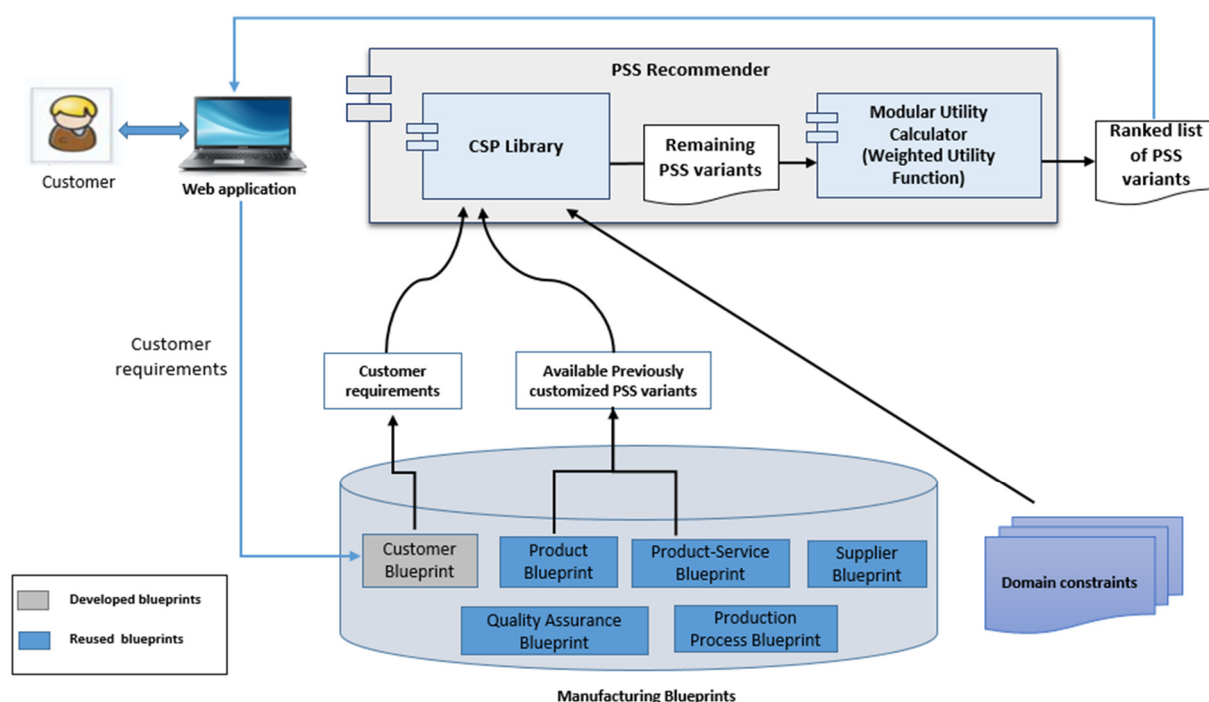


Figure 3. The proposed recommender system architecture.

The architecture consists of a set of integrated manufacturing blueprints, a set of domain constraints, and the PSS recommender. We reused the manufacturing blueprint knowledge base implementations provided in our previous work [6] and extended it with

the manufacturing blueprint extensions presented in Section 4. The extended manufacturing blueprints are implemented using the Ontology Web Language (OWL) standard and protégé tool-suite [49]. The list of domain constraints was generated and stored as a data file with the mzn extension, corresponding to MiniZinc model files.

The recommendation scenario proceeds as follows: the customer interacts with the recommender system through a web application to specify her requirements regarding the PSS variant, as shown in Figure 4. Then, the customer's requirements are maintained in the manufacturing blueprints, using the customer blueprint. Moreover, a list of previously customized PSS variants is also maintained in the manufacturing blueprints, using the product and service blueprints. Both customer requirements and the list of previously customized PSSs are fetched by the PSS recommender. This is enabled by Jena, a java library that allows reading and writing from and into ontology files. The PSS recommender integrates a CSP library, which is based on the MiniZinc <https://minizinc.org/> (accessed on 22 July 2021) open-source constraint problem solving software and modular utility calculator component. The CSP library component is used to filter out PSS variants that do not satisfy the customer's requirements. The remaining PSS variants are processed through the modular utility calculator component to be ranked based on their utility. Finally, the customer is presented with a list of ranked PSS variants to choose from.

Please enter your preferences

Select Laser Generator Type: ☒ CO2 ☐ YAG

Select WorkPiece Type: ☒ Fixed ☐ Controllable

Select Laser Power:

Select Laser Speed:

Select Focus Lens:

Select Included Product Service:

Select Your Environment Temperature Nature:

Enter Maximum Product Price

Enter Maximum Service Price

Please enter your interest in these dimensions:

Product Reliability

Product Performance

Figure 4. Recommender system knowledge acquisition user interface.

The recommender system prototype is implemented as a Maven web application, using Eclipse version (4.10.0). Tomcat server version 8.0 was used. Java, HTML, and JSP were used to handle the front-end and back-end synchronization.

5.4. Discussion

In this paper, we proposed a hybrid knowledge-based recommendation approach for PSS customization recommendations. The majority of previous research studies have focused on providing personalized recommendations in a variety of domains, such as e-commerce, tourism, financial services, etc. However, only a few studies have considered exploring approaches for PSS recommendations. In this sub-section, we compare our proposed recommendation approach to the work presented in [42], due to its high similarity to our proposed approach. The comparison's main criteria are: (i) the ability to deal with data sparsity problems, (ii) the ability to deal with the cold start problem, and (iii) PSS recommendation criteria (i.e., aspects that were taken into account when recommending PSS solutions).

The comparison results show that our approach is free of the cold start and data sparsity issues as it generates recommendations based on the domain knowledge and explicit customer requirements. The approach proposed in [42] does not address the data sparsity and the cold start problems. The approach proposed in [42] only considers the features of the product's associated services, such as service response time, service cost, etc., when recommending PSS solutions. However, our proposed approach takes into account the functional, structural, environmental, quality, and cost aspects of both the product and its associated services when generating PSS recommendations. A summary of the comparison results is provided in Table 13.

Table 13. Comparison between our proposed approach and the approach presented in [42].

Point of Comparison	Our Proposed Approach	Approach in [42]
Dealing with cold start problem	Yes	No
Deal with data sparsity problem	Yes	No
PSS recommendation criteria	Functional, structural, environmental, quality, and cost aspects of both the product and its associated services	Only quality and cost aspects of the product's associated services

To the best of our knowledge, our proposed approach is the first attempt to generate customized PSS recommendations, while taking into account the functional, structural, environmental, cost, and quality aspects of both the product and its associated services.

6. Conclusions

Product-service system (PSS) customization is the ability to offer customized PSSs to satisfy individual customer needs with near-mass-production efficiency. In the context of the PSS customization environment, there can be a multitude of previously customized PSS variants with varying characteristics in order to meet the various needs of different customers. In this paper, we proposed a hybrid knowledge-based recommender system that assists in customers' decisions related to the selection of previously customized PSS variants from a wide range of available ones. This is accomplished by modelling the problem of selecting previously customized PSS variants as a constraint satisfaction problem (CSP) to filter out PSS variants that do not satisfy customers' needs, and then applying a weighted utility function to rank the remaining list of previously customized PSSs based on their utility to the customer.

To the best of our knowledge, our proposed approach constitutes the first attempt to generate personalized PSS recommendations, while taking into account the functional, structural, environmental, cost, and quality aspects of both the product and its associated services. Moreover, the proposed knowledge-based recommendation approach tackles the

cold start problem, as it generates recommendations based on the domain knowledge and explicit customer requirements elicited during the recommendation process. In addition, the generated recommendations are more trustworthy as the domain knowledge is free of noise.

However, our proposed approach suffers from the knowledge acquisition bottleneck as it depends on knowledge engineers, who are required to transform the domain knowledge (item properties and their corresponding constraints) into formal representations. Those knowledge engineers may not always be available, and the knowledge acquisition process may become more complicated when many constraints need to be defined. To overcome the above limitations, we plan to rely on human computation concepts to engage domain experts in the development of knowledge bases. This can be achieved by replacing the complex tasks of knowledge engineers with simple micro-tasks [44,45] that can be completed easily, even by domain experts without technical expertise.

Future research efforts are continuing in parallel and complementary directions. Possible future directions for the consequent work related to the proposals of this paper are (i) visualizing PSS recommendations in a user-friendly manner, which may include the use of domain-specific languages and 3D visualization; (ii) explaining why each of the recommended PSS variants are recommended, allowing the customer to make a more informed decision; (iii) evaluating the proposed RS using a set of evaluation metrics, such as *effectiveness* (i.e., whether the recommendations reflect what the customers want), *utility* (i.e., whether the recommendations brought value to the customer), and *persuasiveness* (i.e., whether the recommendations can change the behavior of customers). Another possible research direction will be focused on realizing the other recommendation facilities identified for the other processes in the PSS customization lifecycle introduced in [14].

Author Contributions: Conceptualization, L.E.; methodology, L.E.; software, L.E.; validation, L.E.; formal analysis, L.E.; investigation, L.E., A.E., and I.M.A.H.; resources, L.E., A.E., and I.M.A.H.; data curation, L.E.; writing—original draft preparation, L.E.; writing—review and editing, L.E., A.E., and I.M.A.H.; supervision, M.E.E.-S., A.E., and I.M.A.H. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Data Availability Statement: Data is contained within the article.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Elgammal, A.; Papazoglou, M.; Krämer, B.; Constantinescu, C. Design for Customization: A New Paradigm for Product-Service System Development. *Procedia CIRP* **2017**, *64*, 345–350. [\[CrossRef\]](#)
2. Song, W. Requirement management for product-service systems: Status review and future trends. *Comput. Ind.* **2017**, *85*, 11–22. [\[CrossRef\]](#)
3. Papazoglou, M.; Elgammal, A.; Krämer, B. Collaborative on-demand Product-Service Systems Customization lifecycle. *CIRP J. Manuf. Sci. Technol.* **2018**, *29*, 205–219. [\[CrossRef\]](#)
4. Bustinza, O.F.; Bigdeli, A.Z.; Baines, T.; Elliot, C. Servitization and Competitive Advantage: The Importance of Organizational Structure and Value Chain Position. *Res. Manag.* **2015**, *58*, 53–60. [\[CrossRef\]](#)
5. Papazoglou, M.P.; Van Den Heuvel, W.J.; Mascolo, J.E. A reference architecture and knowledge-based structures for smart manufacturing networks. *IEEE Softw* **2015**, *32*, 61–69. [\[CrossRef\]](#)
6. Papazoglou, M.; Elgammal, A. The manufacturing blueprint environment: Bringing intelligence into manufacturing. In Proceedings of the 2017 International Conference on Engineering, Technology and Innovation (ICE/ITMC), Madeira, Portugal, 27–29 June 2017; pp. 750–759. [\[CrossRef\]](#)
7. Lu, J.; Wu, D.; Mao, M.; Wang, W.; Zhang, G. Recommender System Application Developments: A survey. *Decis. Support Syst.* **2015**, *74*, 1–38. [\[CrossRef\]](#)
8. Priyanka, P. A Survey Paper on Various Algorithm's based Recommender System. *IOSR J. Comput. Eng.* **2017**, *19*, 27–32. [\[CrossRef\]](#)
9. Bouraga, S.; Jureta, I.; Faulkner, S.; Herssens, C. Knowledge-based recommendation systems: A survey. *Int. J. Intell. Inf. Technol.* **2014**, *10*, 1–19. [\[CrossRef\]](#)

10. Chicaiza, J.; Valdiviezo-Diaz, P. A comprehensive survey of knowledge graph-based recommender systems: Technologies, development, and contributions. *Information* **2021**, *12*, 232. [\[CrossRef\]](#)
11. Beheshti, A.; Yakhchi, S.; Mousaeirad, S.; Ghafari, S.M.; Goluguri, S.R.; Edrisi, M.A. Towards cognitive recommender systems. *Algorithms* **2020**, *13*, 176. [\[CrossRef\]](#)
12. Sulikowski, P.; Zdziebko, T. Horizontal vs. Vertical recommendation zones evaluation using behavior tracking. *Appl. Sci.* **2021**, *11*, 56. [\[CrossRef\]](#)
13. Sulikowski, P. Evaluation of Varying Visual Intensity and Position of a Recommendation in a Recommending Interface Towards Reducing Habituation and Improving Sales. In *Advances in E-Business Engineering for Ubiquitous Computing*; Springer: Cham, Switzerland, 2020.
14. Esheiba, L.; Elgammal, A.; El-Sharkawi, M.E. Recommendation framework for on-demand smart product customization. In *Proceedings of the 21st International Conference on Enterprise Information System, Crete, Greece, 3–5 May 2019; Volume 2*, pp. 177–187. [\[CrossRef\]](#)
15. Edward, T. *Foundations of Constraint Satisfaction*; Academic Press Limited: London, UK; Sandiego, CA, USA, 1993. [\[CrossRef\]](#)
16. Guarino, N.; Oberle, D. What IS an Ontology? *Handb. Ontol.* **2009**, 1–17. [\[CrossRef\]](#)
17. Noy, N.F.; McGuinness, D.L. Ontology Development 101, A Guide to Creating Your First Ontology. *Artif. Intell. Med.* **2001**. [\[CrossRef\]](#)
18. Vegetti, M.; Leone, H.; Henning, G. PRONTO: An ontology for comprehensive and consistent representation of product information. *Eng. Appl. Artif. Intell.* **2011**, *24*, 1305–1327. [\[CrossRef\]](#)
19. Panetto, H.; Dassisti, M.; Tursi, A. ONTO-PDM: Product-driven ONTOlogy for Product Data Management interoperability within manufacturing process environment. *Adv. Eng. Inform.* **2012**, *26*, 334–348. [\[CrossRef\]](#)
20. Shen, J.; Wang, L. A new perspective and representation of service. *Int. Conf. Wirel. Commun. Netw. Mob. Comput. WiCOM* **2007**, 3176–3179. [\[CrossRef\]](#)
21. Doultsinou, A.; Roy, R.; Baxter, D.; Mann, A. Developing a service knowledge reuse framework for engineering design. *J. Eng. Des.* **2009**, 37–41. [\[CrossRef\]](#)
22. Roy, R.; Xu, Y.; Annamalai, B.G.; Hussain, R.; Cakkol, M.; Roy, R. AN ONTOLOGY FOR PRODUCT-SERVICE SYSTEMS. *J. Eng. Des.* **2011**. [\[CrossRef\]](#)
23. Hajimohammadi, A.; Cavalcante, J.; Gzara, L. Ontology for the PSS Lifecycle Management. *Procedia CIRP* **2017**, *64*, 151–156. [\[CrossRef\]](#)
24. Shen, J.; Wang, L.; Sun, Y. Configuration of product extension services in servitisation using an ontology-based approach. *Int. J. Prod. Res.* **2012**, 7543. [\[CrossRef\]](#)
25. Yang, D.; Dong, M.; Miao, R. Development of a product configuration system with an ontology-based approach. *Comput. Aided. Des.* **2008**, *40*, 863–878. [\[CrossRef\]](#)
26. Felfernig, A.; Friedrich, G.; Jannach, D.; Zanker, M. Developing Constraint-based Recommenders. In *Recommender Systems Handbook*; Springer: Berlin/Heidelberg, Germany, 2011; pp. 1–29.
27. RICHTER, M.; AAMODT, A. Case-based reasoning foundations. *Knowl. Eng. Rev.* **2006**, *20*, 203–207. [\[CrossRef\]](#)
28. Lam, X.N.; Vu, T.; Le, T.D.; Duong, A.D. Addressing Cold-Start Problem in Recommendation Systems. In *ICUIMC '08: Proceedings of the 2nd International Conference on Ubiquitous Information Management and Communication*; Association for Computing Machinery: New York, NY, USA, 2008; pp. 208–211.
29. Yao, X.; Moon, S.K.; Bi, G. A hybrid machine learning approach for additive manufacturing design feature recommendation. *Rapid Prototyp. J.* **2017**, *23*, 983–997. [\[CrossRef\]](#)
30. Zhang, W.Y.; Zhang, S.; Chen, Y.G.; Pan, X.W. Combining social network and collaborative filtering for personalised manufacturing service recommendation. *Int. J. Prod. Res.* **2013**, *51*, 6702–6719. [\[CrossRef\]](#)
31. Liu, J.; Chen, Y. A personalized clustering-based and reliable trust-aware QoS prediction approach for cloud service recommendation in cloud manufacturing. *Knowledge-Based Syst.* **2019**, *174*, 43–56. [\[CrossRef\]](#)
32. Hao, Y.; Fan, Y.; Zhang, J. Service recommendation based on description reconstruction in cloud manufacturing. *Int. J. Comput. Integr. Manuf.* **2019**, *32*, 294–306. [\[CrossRef\]](#)
33. Simeone, A.; Caggiano, A.; Deng, B.; Boun, L. A deep learning based-decision support tool for solution recommendation in cloud manufacturing platforms. *Procedia CIRP* **2019**, *86*, 68–73. [\[CrossRef\]](#)
34. Zhang, Z.; Lin, H.; Liu, K.; Wu, D.; Zhang, G.; Lu, J. A hybrid fuzzy-based personalized recommender system for telecom products/services. *Inf. Sci.* **2013**, *235*, 117–129. [\[CrossRef\]](#)
35. Zanker, M.; Aschinger, M.; Jessenitschnig, M. Constraint-based personalized configuring of product and service bundles. *Int. J. Mass Cust.* **2010**, *3*, 1–19.
36. Jannach, D.; Zanker, M.; Fuchs, M. CONSTRAINT-BASED RECOMMENDATION IN TOURISM: A MULTIPERSPECTIVE CASE STUDY. *Inf. Technol. Tour* **2009**, *11*, 139–155. [\[CrossRef\]](#)
37. Felfernig, A.; Friedrich, G.; Jannach, D.; Zanker, M. An Integrated Environment for the Development of Knowledge-Based Recommender Applications. *Int. J. Electron. Commer* **2006**, *11*, 109–129. [\[CrossRef\]](#)
38. Felfernig, A.; Isak, K.; Szabo, K.; Zachar, P. The VITA financial services sales support environment. *Proc. Natl. Conf. Artif. Intell.* **2007**, *2*, 1692–1699.

39. Dhelim, S.; Ning, H.; Aung, N.; Huang, R.; Ma, J. Personality-Aware Product Recommendation System Based on User Interests Mining and Metapath Discovery. *IEEE Trans. Comput. Soc. Syst.* **2020**, *7*, 86–98. [\[CrossRef\]](#)
40. Deepak, G.; Kasaraneni, D. Ontocommerce: An ontology focused semantic framework for personalised product recommendation for user targeted e-commerce. *Int. J. Comput. Aided. Eng. Technol.* **2019**, *11*, 449–466. [\[CrossRef\]](#)
41. SINGH, M.; RISHI, O.P. Event driven Recommendation System for E-commerce using Knowledge based Collaborative Filtering Technique. *Scalable Comput. Pract. Exp.* **2020**, *301*, 2014–2016. [\[CrossRef\]](#)
42. Song, W.; Sakao, T. An environmentally conscious PSS recommendation method based on users' vague ratings: A rough multi-criteria approach. *J. Clean Prod.* **2018**, *172*, 1592–1606. [\[CrossRef\]](#)
43. Song, W. Personalized Recommendation of Customizable PSS to Customers. In *Customization-Oriented Design of Product-Service System*; Springer Nature: Singapore, 2018; pp. 1–202. [\[CrossRef\]](#)
44. Felfernig, A.; Haas, S.; Ninaus, G.; Schwarz, M.; Ulz, T.; Stettinger, M. RecTurk: Constraint-based Recommendation based on Human Computation. *ACM Recomm. Syst.* **2014**, *1*, 1–6.
45. Frankowski, D.; Lam, S.K.; Sen, S.; Harper, F.M.; Yilek, S.; Cassano, M. Recommenders Everywhere: The WikiLens Community-Maintained Recommender System. *wikiSym* **2007**, *7*, 47–60. [\[CrossRef\]](#)
46. Freuder, E.C.; Mackworth, A.K. Constraint Satisfaction: An Emerging Paradigm. In *Foundations of Artificial Intelligence*; Elsevier: Amsterdam, The Netherlands, 2006; Volume 2. [\[CrossRef\]](#)
47. Von Winterfeldt, D.; Edwards, W. *Decision Analysis and Behavioral Research 1986*; Cambridge University Press: Cambridge, UK, 1986.
48. Witten, H.I.; Frank, E.; Mark, A.; Hall, C.J.P. *Extending Instance-Based and Linear Models*; Elsevier: Amsterdam, The Netherlands, 2017. [\[CrossRef\]](#)
49. BMIR. Protégé n.d. Available online: <https://protege.stanford.edu/> (accessed on 7 January 2021).