



HAL
open science

A design pattern for industrial robot: User-customized configuration engineering

Jing Li, Joanna Daaboul, Shurong Tong, Magali Bosch-Mauchand, Benoît Eynard

► **To cite this version:**

Jing Li, Joanna Daaboul, Shurong Tong, Magali Bosch-Mauchand, Benoît Eynard. A design pattern for industrial robot: User-customized configuration engineering. *Robotics and Computer-Integrated Manufacturing*, 2015, 31, pp.30-39. <10.1016/j.rcim.2014.06.005>. <hal-01076064>

HAL Id: hal-01076064

<https://hal.science/hal-01076064v1>

Submitted on 1 Apr 2023

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.



Distributed under a Creative Commons CC BY-NC 4.0 - Attribution - Non-commercial use - International License

A design pattern for industrial robot: User-customized configuration engineering

Jing Li ^{a,b}, Joanna Daaboul ^a, Shurong Tong ^b, Magali Bosch-Mauchand ^a, Benoît Eynard ^{a,*}

^a Université de Technologie de Compiègne, Department of Mechanical Systems Engineering, CNRS UMR7337 Roberval, CS 60319, 60203 Compiègne Cedex, France
^b Institute for Design Management, Northwestern Polytechnical University, Xi'an, Shaanxi, PR China

According to the characteristics of industrial robots, user-customized configuration design pattern is showing to achieve rapid development of industrial robotics. In this paper, the definition of user-customized configuration design pattern is determined. The implementation approach of this pattern is given through introducing what role industrial robot stakeholders including suppliers, manufacturers, designers and users should play in the pattern. Then, the key technologies of implementation approach are introduced. Finally, system dynamics models are established for this design pattern and for traditional design pattern of industrial robots and are simulated by Anylogic simulation software. The simulation results prove that the proposed design pattern has better performances than traditional design pattern with regard to inventory and order response speed.

1. Introduction

Due to the development of manufacturing automation and flexible manufacturing systems, the demand for various types of industrial robots is increasing. Industrial robot is a type of mechatronic products containing multiple field components such as machinery, electronics, control unit, and computer. It is characterized by complex structure, long lead times, and high manufacturing costs. However, an industrial robot is required to achieve high degree of customization for finishing special work. These characteristics are constrained and limit the rapid development of industrial robot.

In accordance with this problem, dozens of studies have been carried out. These studies could be divided into two categories. The first one is to design heterogeneous reconfigurable robot. The robot architecture is designed to be modular [1,2], and a new robot configuration can be obtained by reconfiguring modules when new requirement emerges [3,4]. The second one is to study control system reconfiguration based on physical modules reconfiguration [5,6]. The Recrob of Technical University of Cluj-Napoca [7], SMART of Universidad Politécnica de Madrid [8], and Odin of University of Southern Denmark [9] are all heterogeneous robots based on reconfigurable theory, and these robots can be quickly reconfigured based on different tasks.

The above research solves the problem of the industrial robot development speed from different techniques perspective. But only

realizing reconfigurable modular industrial robot from technical view is not sufficient to significantly reduce the trade-off between customization and delivery time [10]. The product is not designed by customers and thus cannot meet every customer's need. Pacheco et al. made the attempt at customers' configuring the robot by themselves [11]. They developed a heterogeneous toy robot platform on which non-professionals were able to assemble modules into different robots. But they focused on the customizable realization of toy and service robot, and little research was observed on the customizable realization of industrial robot. Therefore, this paper puts forward a realization approach by which customers may configure required industrial robot by themselves. This approach provides a supplementary research to previous studies from management view, thereby solving the contradiction between an industrial robot highly customized and a faster response.

Changing the industrial robot development approach from professional designer design to customer configuration design is not a simple task. It requires modifying the respective roles of relevant enterprises such as industrial robots manufacturers, parts providers and developers in industry sector. To ensure that customers can design industrial robot by themselves, a clearly specified set of processes is needed to manage all types of stakeholders. In the following section, the user-customized configuration design pattern definition and implementation steps are introduced. In Section 3, the proposed framework as well as the process of defining this pattern, including needed key technologies, is addressed. Section 4 presents a comparison of the proposed design pattern and traditional design pattern via simulation.

* Corresponding author.

E-mail address: benoit.eynard@utc.fr (B. Eynard).

Finally, Section 5 concludes on the main obtained results and presents key ideas for future work.

2. Proposal: user-customized configuration design pattern

2.1. Definition of user-customized configuration design pattern

According to the state of the art presented in Section 1, it can be stated that industrial robot is nowadays a generally heterogeneous reconfigurable system. The customer is responsible for proposing requirements, and the designer is responsible for reconfigurable industrial robot designing based on customer's requirements. When new demands arrive, based on previous reconfigurable industrial robot, designer can quickly configure a new industrial robot with little redesign. Nevertheless, in this process, the iterations still exist between customers and designers. As mentioned above, industrial robot is required to achieve high degree of customization since it is required to achieve special work, and it is impossible for designers to meet customers' requirements accurately even after several iterations. When the performance of the product designed by designers does not entirely meet customer's expectations, the designers need to modify their design to better meet customers' requirements. This process has to be repeated a few times before the design of an industrial robot is achieved. Even though reconfigurable product accelerates the design, these iterations will still lead to increased cost and order fulfillment lead time. In order to overcome this problem, a new design pattern for industrial robot is proposed: user-customized configuration design pattern.

Before giving the definition of user-customized configuration design pattern, the concept of pattern should be clarified. A classic definition was given by Alexander et al.: Each pattern describes a problem which occurs over and over again in our environment, and then describes the core of the solution to that problem, in such a way that you can use this solution a million times over, without ever doing it the same way twice [3]. A design pattern is a solution to a recurring problem [12]. In general, design pattern is used in software engineering to solve problems in code design. There are two key elements in the concept of design pattern: problem and solution. Based on this, user-customized configuration design pattern is defined as follows.

Definition 1. (User-Customized Configuration Design Pattern):

In the process of industrial robot configuration design, [Problem] the rework of design often arises to increase cost and order fulfillment lead time because of not meeting customer's requirement completely. To solve this problem, [Solution] product users are allowed to customize the product by themselves. This solution to solve the problem is called user-customized configuration design pattern. In this pattern, the final configuration is not completed by the designer but by the user according to his/her own needs. In this case, the designer provides basic components, and the user configures the product using these components.

Industrial robots are usually used in factories to do simple and repetitive work, thus generally a single function is required. In addition, the users of industrial robots are usually factories that have employees with mechanical expertise. Therefore the implementation of user-customized configuration design is possible in such a case.

2.2. User-customized configuration design pattern implementation method

From Definition 1, we know that user-customized configuration design pattern is a guidance method. The question is how to

implement this pattern? This paper presents the implementation steps of user-customized configuration design pattern as follows:

- (1) Customers put forward demand.
- (2) Product developers design industrial robot components according to the demand.
- (3) Component providers choose components that are suitable for their own production conditions to produce.
- (4) Customers/users customize industrial robot through selecting components by their own, or adopt a solution previously designed by other users. Users can also publish new requirements that cannot be met by the existing components.
- (5) According to the design results, product providers select the optimal combination of component providers, make components purchasing plan, and then deliver the purchased components to product operator. The components also can be assembled to industrial robot by the product providers before sending them to product operators.
- (6) Product operators deliver the components or finished products to customers after choosing appropriate logistics solution.
- (7) Quality supervision departments supervise the various products and service quality of the participants, and ensure that the entire process and the final products meet national standards.

In the implementation, there are several concepts that must be clarified. The first one is the concept of Component. In mechanical field, a component refers to an element that is composed of one or more parts. Based on the concept of component, this paper defined a concept of I-Component. If not otherwise specified, when discussing problems related to user-customized configuration design, the component mentioned in this paper refers to component related to the concept of I-Component.

Definition 2. (I-Component): An I-Component is a component that encapsulates function realization and internal structure, and can enter into assembly phase directly on the premise of meeting interface criteria in a certain field.

The paper uses "Cpt" to represent I-Component, and $Cpt = \{St, Int\}$. Wherein, "St" represents internal physical structure of the I-Component. A tuple formula is used to express "St": $St = \{sc, r\}$, where "sc" is the sub-I-Component set of the I-Component, "r" is the composition relationship of the sub-I-Component. When the I-Component is non-degradable, the values of "sc" and "r" are zero. "Int" represents I-Component interface, and $Int = \sum_{i=1}^n ip_i$, where $i=1, 2, \dots, n$, "n" is the I-Component interface number. The " ip_i " represents interface parameters of number "i". Interface parameters include interface size, interface location, communication standard, etc.

The definition could be understood from the following points:

- (1) I-Component is a physical entity whose interface meets certain criteria.
- (2) I-Component interface and structure are divided. I-Component is a black-box for its consumers; it exchanges energy, matter and information with the outside world only by interfaces.
- (3) I-Component consumers assemble I-Components into products through a simple way similar to "building blocks".

The concept of I-Component is given to increase the diversity and convenience of the user optional components. The industrial robot of manufacturing companies, such as FANUC and YASKAWA of Japan, American Robot of America, ABB of Sweden, KUKA of Germany, has been realized in a reconfigurable modular structure. But due to the non-unification of the upper standard and due to technical security in companies, industrial robots can only achieve

modules reuse within the same company, and cannot achieve matching across different companies. This factor limits the user's configuration design. The I-Component provides a standard interface, and encapsulates component structure that the companies are highly sensitive to but customer is not concerned about. Companies can implement the same interface by different techniques. In this way, companies can release components on the premise of maintaining core technology. Component encapsulation also blocks technical view for the users, which facilitates for the users to configure products.

It is important to note that the concept of module and I-Component can be easily confused. In the research of mechatronics product modularization, a module refers to the physical realization of a function [13]. The module focuses on dividing the components of the product into several parts (which is called module) according to the dependencies between components and each module has an independent function. For example, in reconfigurable modular robot, a module usually contains processors, motors, transmission devices, etc., and the module can be tested and run independently [14,15]. In the research of user-customized configuration design in this paper, a single I-Component does not necessarily realize one independent function. It focuses on encapsulating component, and the interaction between I-Components is achieved through standardized interfaces only.

In fact, the concept of module and I-Component is not completely different. With the continuous development of product modularization research, the module interface also needs to be standardized, which is identical to the requirement on I-Component. For highly modular product, a module of the product can become a plug-and-play I-Component by being encapsulated. On the other hand, the idea of modularization can be introduced into the design of I-Component. The I-Components having more interaction could be designed as a module, and then encapsulate this module to a big I-Component. Doing this could reduce the workload of configuration design.

A second concept that needs to be clarified is the product concept. Product refers to configurable industrial robot designed in user-customized configuration design pattern. A product is configurable when each individual product is adapted to the needs of a particular customer order [8]. A configurable product is also defined as a product that is composed according to the demands of a particular customer order based on a generic product structure [16]. The main feature of a configurable product is its modular structure, and the possibility to create individual products by selecting different predefined components [17]. The configurable product is generally composed by common components and special components. Common components are the components required in different products, such as the wrist, the drive system, etc. Special components are used to achieve a specific function for different users, such as the various sensors, end-effectors, etc. In industrial robots design, a design template can be built by distinguishing different component types, which can improve the configuration efficiency. Based on a comprehensive collection of existing research, the concept of configurable product is defined as follows.

Definition 3. (Configurable Product): Use $ComCpt_i$ to represent common components and $SpeCpt_i$ to represent special components. Set $\sum_{i=1}^m ComCpt_i$ as the collection of $ComCpt_i$, and $\sum_{i=1}^n SpeCpt_i$ as the collection of $SpeCpt_i$. Product is represented by \mathbf{P} , and

$$\mathbf{P} = C \left\{ \sum_{i=1}^x ComCpt_i, \sum_{i=1}^y SpeCpt_i \right\}.$$

wherein, x , y , m , and n are integers,

$$\sum_{i=1}^x ComCpt_i \subseteq \sum_{i=1}^m ComCpt_i, \quad \sum_{i=1}^y SpeCpt_i \subseteq \sum_{i=1}^n SpeCpt_i.$$

\mathbf{C} refers to constraints. The constraints here refer to the rules that must be followed to guaranty the product feasibility when assembling components to product. For example, when one gear meshes with another, the required transmission ratio determines the number of gear teeth; when a sensor is connected with a controller, not only the interfaces but also the communication protocol need to be consistent, etc.

Then product \mathbf{P} is called the configurable product, and the collection composed of \mathbf{P} is called the configurable product family.

The third concept needed to be clarified is user. In user-customized configuration design, the user is defined as follows.

Definition 4. (User): User refers to all people that can make design activities. There are two kinds of users: the enterprise or individual that uses the product directly (i.e. customer); the enterprise or individual that provides product solution for product end-user.

The paper uses “**Us**” to present user.

The fourth concept needing to be clarified is user-customized design. Its definition is as follows.

Definition 5. (User-Customized Design): Under certain constraint conditions, user assembles components selected from the component library composed by configurable product family, and obtains product design results satisfying his/her own needs. The meaning of constraint conditions here is the same with \mathbf{C} in Definition 3.

User-customized design can be formally represented as follows:

$$Us + \left\{ \sum_{i=1}^m ComCpt_i \cup \sum_{i=1}^n SpeCpt_i \right\} \xrightarrow{C_{(Select)}} P$$

wherein $C_{(Select)}$ represents an act of selecting needed components under constraints.

Finally and most importantly, from the implementation steps of user-customized configuration design pattern we know that each step does not exist in isolation, and the implementation of each step needs the results of the previous step as an input; therefore, a mechanism supporting information sharing and communication between stakeholders is needed. A platform is usually used to achieve a synergy of sales force automation, product design, manufacturing planning, and supply chain management within a consistent framework [18]. To make the implementation steps smoothly, this paper defines a design platform as follows.

Definition 6. (Design Platform): The design platform is formed of hardware and software environments and technical processes supporting user-customized configuration design pattern realization.

The platform realizes user-customized configuration design pattern, thus it must be open. The openness is embodied in three aspects: the first is that people using the platform are diversified, hence the platform must be able to be used by all stakeholders such as users, product developers, component providers, product operators, etc.; the second is that the enterprises releasing components on the platform and the enterprises using components via the platform are diversified, thus the platform must provide component release criterion, and components meeting the criterion can enter the platform; the third is that the user does not only design product by himself, but may also choose designs previously realized by other users, therefore the platform must support all users design product and permit sharing design results.

3. Key technologies of user-customized configuration design pattern

In Section 2, a design platform is put forward to realize user-customized configuration design pattern, and stakeholders can execute a variety of operations such as design, sale, purchase, service, supervision and other activities on the platform. According to UML principles applied for product structuring and modeling [19], Fig. 1 depicts a use case diagram of the platform, which shows the different stakeholders' activities on the platform in detail. Product developers receive customers' needs and release new component design results. Component providers release produced components. Users design the product, and as customers, they can also put forward new demand and track their order after purchasing the product. Product providers manage components providers, obtain configuration result and then select the optimal combination of component suppliers for every product. Product operators provide the optimal logistics solution, and release the order status. Quality supervision departments supervise the various products and service quality of the participants. Website service providers provide background maintenance, and resolve the technical problems encountered by the participants in the process of using platform.

Based on the open design platform use case diagram, technology process of open user-customized configuration design platform is proposed in Fig. 2. Product developers store the developed industrial robot components to the component prototype library. At the same time, the constraints and algorithm when configuring components to comply with will be stored to configuration rules and algorithms library. Component providers extract design information from component prototype library, and then release the information of components which have been produced such as type, specification, producing area, price etc. to the component library. Users configure industrial robot through customized template library by calling the data in component prototype library and configuration rules and algorithms library. Users also can publish new requirements that cannot be met by the existing components to the template library. Through reading configuration result from template library, product providers make components purchasing plan. Product operators store the logistics information into the logistics information database for the user query. Quality supervision departments interact with all participants, supervise the various products and service quality of the participants, and ensure the entire process and the final products meet national standards. Website service providers also interact with all stakeholders; they provide background maintenance, and

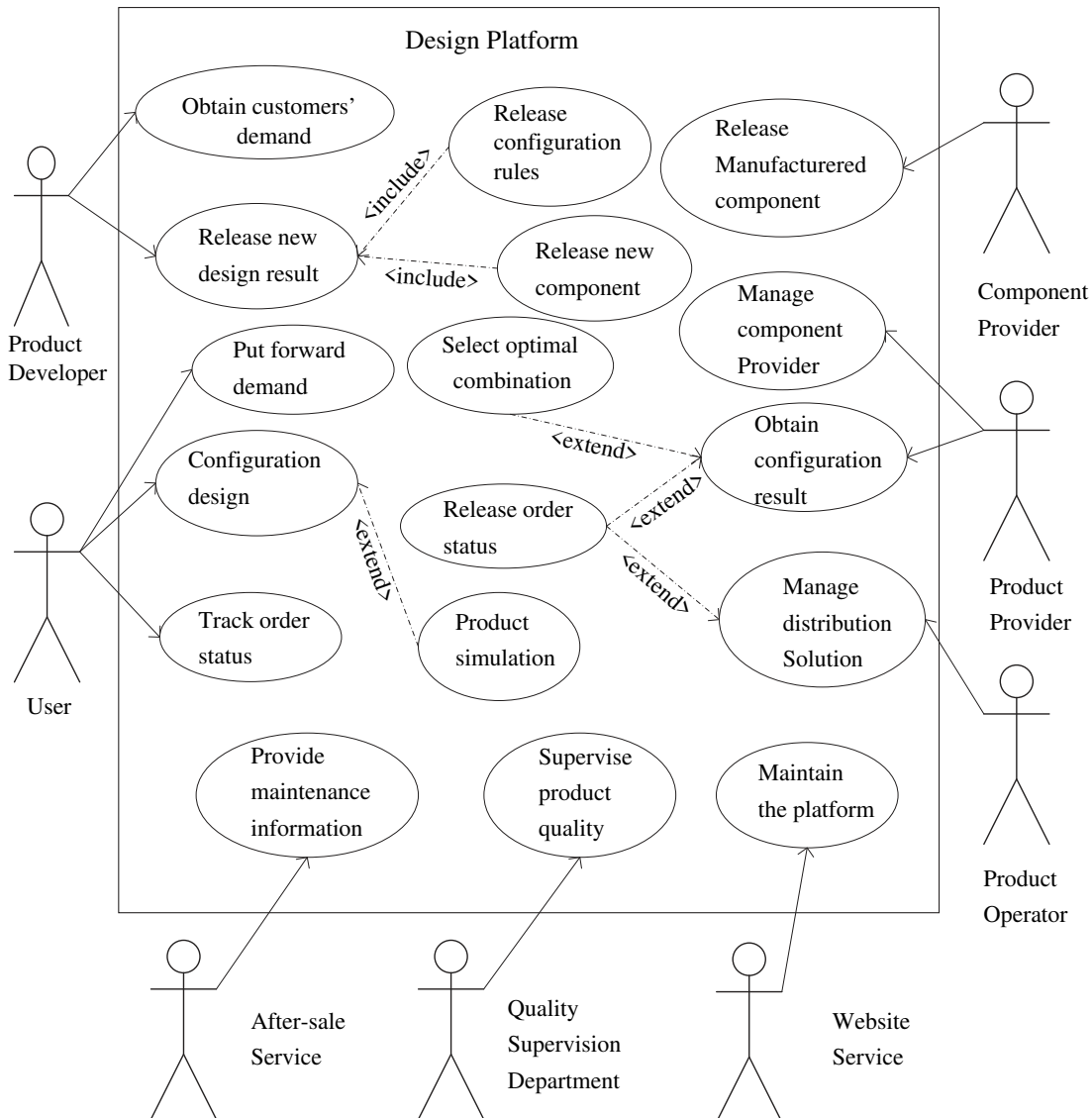
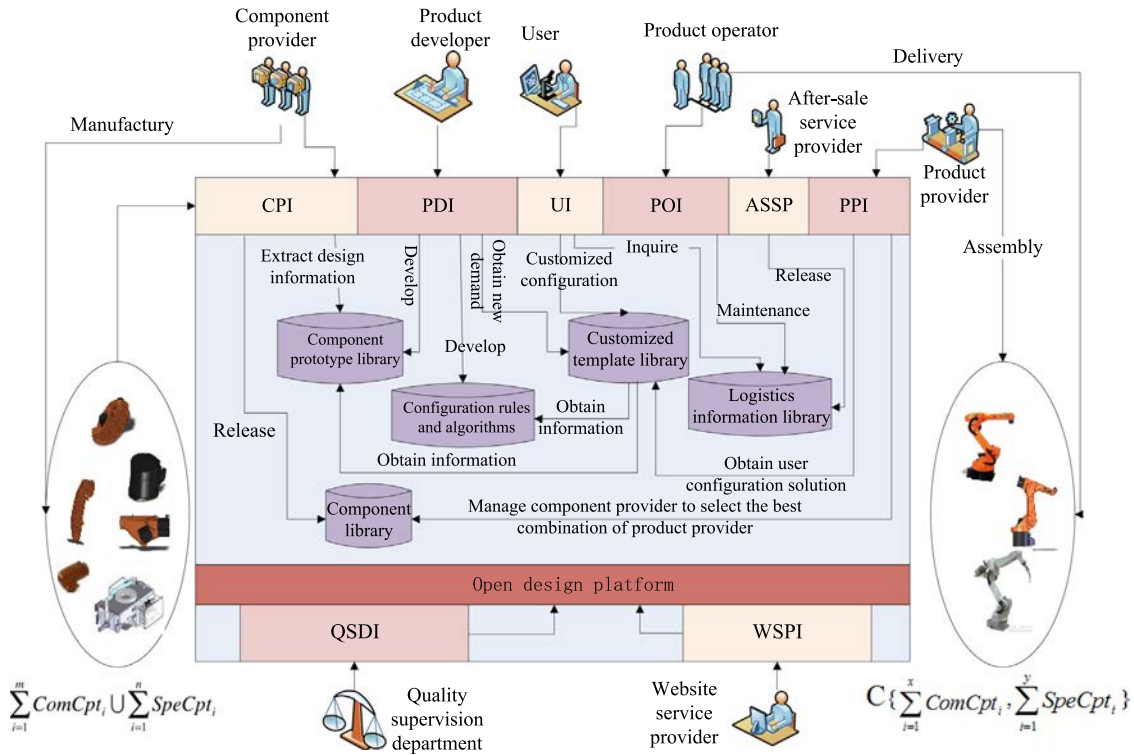


Fig. 1. Use case diagram of open design platform.



CPI: Component provider interface; PDI: Product developer interface; UI: User interface; POI: Product operator interface; ASSPI: After-sale service provider; PPI: Product provider interface; QSDI: Quality supervision department interface; WSPI: Website service provider interface

Fig. 2. Technology process of design platform.

resolve the technical problems encountered by the participants while using the platform.

There are some key technologies needed in order to achieve this platform. First, the unified expression method of the industrial robot component needs to be determined. The representation of robot modules is the basis for configuration [20,21]. Industrial robot components involve different fields, including mechanical components, electrical components, etc., and it does not only consist of a physical connection between the various components. The connection involves electrical parts, control signals, etc., In order to present all of the components on the same platform and to configure the components of different fields, the unified expression method of the multidisciplinary component needs to be determined.

Second, a user-customized template for industrial robot needs to be developed. The platform supports user-customized configuration design, but users are not professional designers, and hence cannot guarantee the manufacturability and rational construction of the product. Therefore, on the basis of components expression, the platform needs to provide a customized template and a product simulation software interface to guide the user in the design activities. A customized template is defined as follows.

Definition 7. (Customized Template): Customized template is a generic and abstract structure of a product. User instantiates this structure via selecting components from a configurable product family, then products with specific features can be obtained.

Third, component release criterion needs to be set. On the platform, all potential users having design capabilities can act as designers and release their own design results. The customized template can accurately call the released components and

component providers can quickly search various components; some criteria need to be satisfied when the components are being released. Component release criterion is defined as follows.

Definition 8. (Component Release Criterion): Component release criterion refers to the technical indicators that need to be provided when the component is released to the platform. The indicators consist of two parts: indicators that uniquely correspond to the functionality of the designed components that component developers release to component prototype library, generally refers to the interface parameters; and indicators that realize the differentiation of the produced components that component suppliers release to component library, including accuracy, materials used, load bearing, price, origin, delivery time, etc.

4. Validation of proposed user-customized configuration design pattern through simulation

Traditional design pattern of industrial robot is engineer-to-order (ETO). In this pattern, every user can obtain a product designed based on their customized needs. The new design pattern proposed in this paper can also meet the needs of every user. At the same time, the new design pattern has obvious advantages in terms of cost and order response speed. In this section, system dynamics model for the two patterns is built to analyze their design and production process. System dynamics is a modeling and simulation method that is widely used in production processes and is very suitable for the simulation of complex dynamic systems [22]. The performance advantage of the proposed model is validated by the simulation analysis.

4.1. Introduction of traditional design pattern and user-customized configuration design pattern

Traditional design method for industrial robot is ETO. The company designs products according to users' requirements, generates orders, and organizes production processes based on the orders. The customer order decoupling point (CODP), which is traditionally defined as the point in the value chain for a product, where the product is linked to a specific customer order [23], is positioned before the design activities. The operation process of the entire traditional design and production process of industrial robot is shown in Fig. 3. Users put forward their demands; the design department of the company designs several products to meet the needs of each user, and then the production department arranges the production plan. The entire production process begins with basic components provider. It produces components and parts in accordance with orders, and promotes the entire production process to downstream until the products are submitted to users.

In user-customized configuration design pattern, the production mode is assemble-to-order (ATO). The running process is shown in Fig. 4. The user selects components from the component

library according to the clarified needs, configures the product on the customized template, and sends his/her order to the company after the completion of configuration design. The company assembles the components to product based on the user's order. Basic components provider produces components according to the components prototype released on the platform. A component library is formed by various providers producing components, and is selected by assemblers freely. The core of the whole process is the assembly stage. Company purchases quantitative standard components in advance as inventory; hence the company can arrange for the assembly activities immediately when they receive customer orders. At the same time, the company continues to purchase components based on the number of orders and inventory.

4.2. System dynamics model of the two design pattern

Guohua and his research team proposed a system dynamics simulation method of different supply chain modes [24]. But they focused on delivery ratio research and did not consider the design pattern influence on order. Based on their research and the analysis of the entire design and production process in

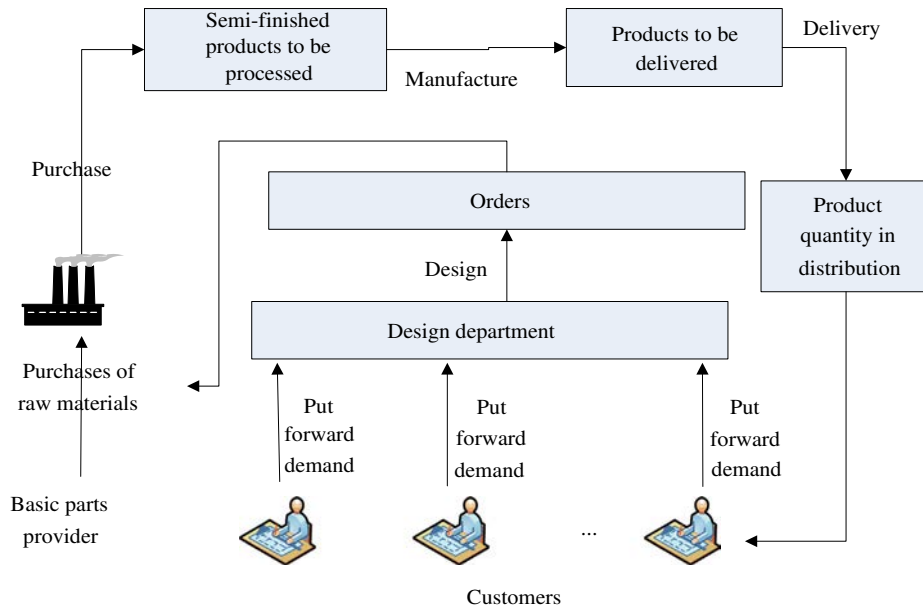


Fig. 3. Traditional design process of industrial robot.

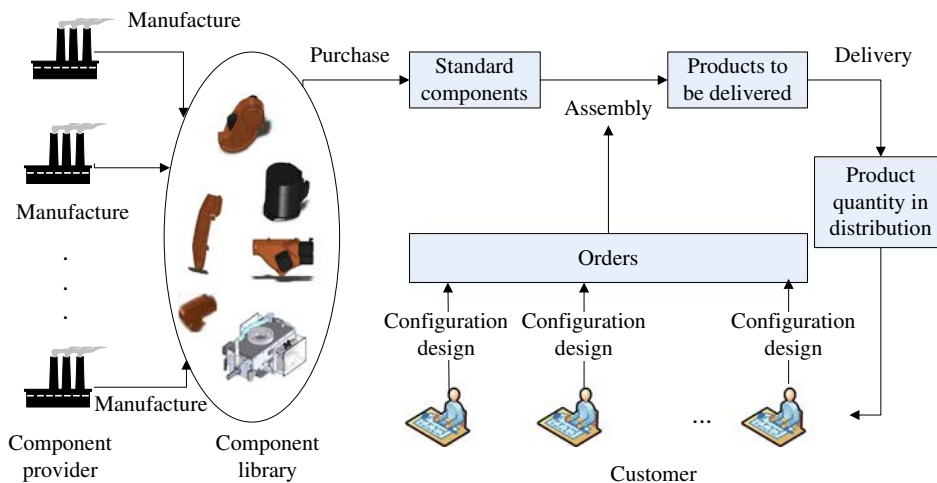


Fig. 4. User-customized configuration design process of industrial robot.

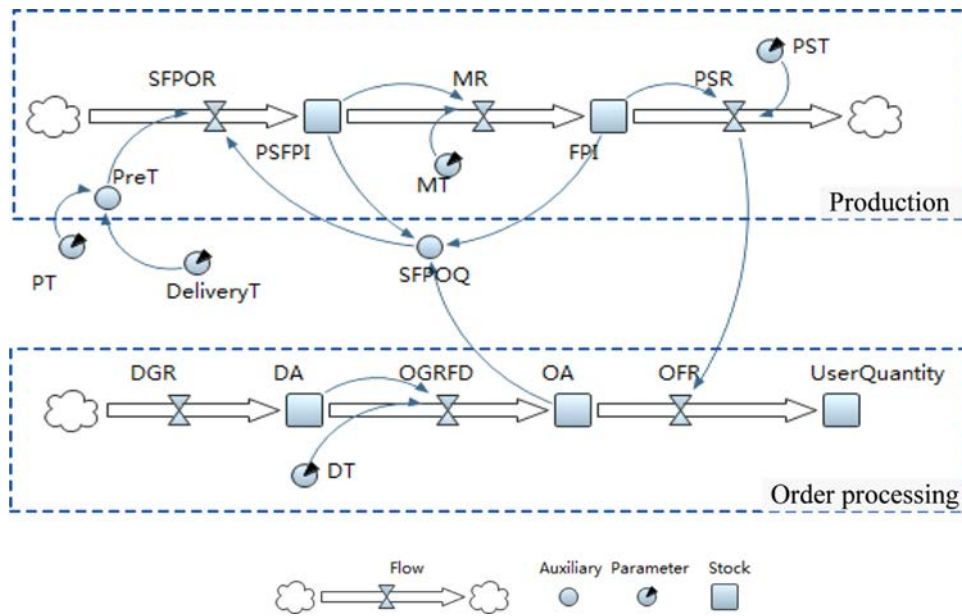


Fig. 5. Traditional design pattern model.

Section 4.1, the simulation models of the two patterns are built using system dynamics on Anylogic simulation platform. In system dynamics model, “stock” is used to describe things increasing or decreasing over time in the real world. In the process of things increasing or decreasing, the “flow” generated, and the difference between inflow and outflow of the stock is the stock value. “Parameter” represents a constant or a variable outside the model which has an impact on the model running. “Auxiliary” represents the internal information of model, and describes the relationship between variables.

The system dynamics model of traditional design mode is shown in Fig. 5. The entire process contains two parts: order processing and product production.

In the step of order processing, new demands emerge continuously in market. It takes time to design products according to demand, thus the stock “Demand Accumulation (DA)” generates in the case of the demand not being met immediately. The value of DA is decided by the difference between its inflow Demand Generated Rate (DGR) and outflow Order Generated Rate From Demand (OGRFD). The company starts to design products after receiving users’ demands, transfer product requirements into orders, and then begins to arrange production according to orders. In this process, the OGRFD is determined by “Demand Accumulation”/“Design Time (DT)”. Order Accumulation (OA) is determined by the difference between its inflow OGRFD and outflow Order Finished Rate (OFR). The OA minus the orders that has been processed (including the orders being processed – “Processing semi-finished products inventory (PSFPI)” and the orders that have been completed – “Finished products inventory (FPI)”) is the required Semi-Finished Products Order Quantity (SFPOQ).

Concerning the production step, semi-finished products need to be purchased after receiving orders from design department, thus there is no initial semi-finished products inventory. It takes some time to transform semi-finished products into finished products, thus the stock “Processing Semi-Finished Products Inventory (PSFPI)” is generated in production process. Similarly, the stock “Finished Products Inventory (FPI)” is generated since it is not possible to deliver all the finished products immediately. The stock “Processing Semi-Finished Products Inventory” is determined by the difference between its inflow Semi-Finished Products Order Rate (SFPOR) and outflow Manufacturing Rate (MR).

Similarly, the stock “Finished Products Inventory” is determined by the difference between its inflow manufacturing rate and outflow Product Shipping Rate (PSR). The product order is finished when the product begins to be shipped, so the product shipping rate is equivalent to the order finished rate.

The system dynamics model of user-customized configuration design pattern is shown in Fig. 6. The process also contains two parts: order processing and product production.

Concerning the order processing step, this pattern also generates stock “Order Accumulation (OA)”, which is similar to the traditional mode. The accumulated orders minus the Finished Products Order (FPO) is the Component Quantity to be Assembled (CQA). Notice, however, that the users design the orders by themselves, and the design time is the time required for each user in order to design his own product. It is different from the first pattern, whose design time is the time needed by the design department in order to design all orders. So, after one user completes his own design, his demand will translate into order.

Concerning the production step, since the components used in design process are selected in the component library, company can purchase a certain amount of components from the component provider in advance as target inventory. On the premise of maintaining normal production, the inventory is as little as possible. In this case, company can immediately plan for the assembly using the inventory of components when the user orders arrive. When the component inventory is less than the target inventory, the components are ordered in time, and the Component Order Quantity (COQ) is determined by the difference between Component Target Inventory (CTI) and Component Inventory (CI). The stock “Component Inventory (CI)” is determined by the difference between its inflow Component Order Rate (COR) and outflow Assembly Rate (AR).

4.3. Simulation results and analysis

Before running the simulation, the model parameters are set. The simulation time unit is set in days, the simulation step is set in 0.2, and the simulation length is set in 80. The generation rate of demand is in line with a normal distribution whose mean is 10 and variance is 10. The random number distribution area of the normal distribution is [0, 20]. At the 30th day, the generation rate of

demand is changed. Its mean is changed from 10 to 15. When the user demand generation rate increases, the components target inventory in the second pattern is modified into 35 by assembly department in order to accommodate the increase of demand, while the assembly time and shipping time are modified to 3 days. Similarly, production time and shipping time will be adjusted to two days in the first pattern when the user demand generation rate increases.

The model parameters setting and variable formula used in Fig. 5 are listed in Tables 1 and 2.

For comparison of the overall performance of the two systems, the time initial value of traditional design pattern model is set to

the same value as user-customized configuration design pattern. The specific settings are listed in Tables 3 and 4.

The resulting order generation rates of the two patterns are shown in Fig. 7a and b. Simulation results for production inventory and orders accumulation are shown in Fig. 8a and b.

The simulation results are based on the following assumptions:

- (1) The time required to design a product in Figs. 3 and 4 is the same. Because the models only simulate the whole process from demand generation to product delivery in the two modes, and do not simulate the specific product design process and design time of the two modes.

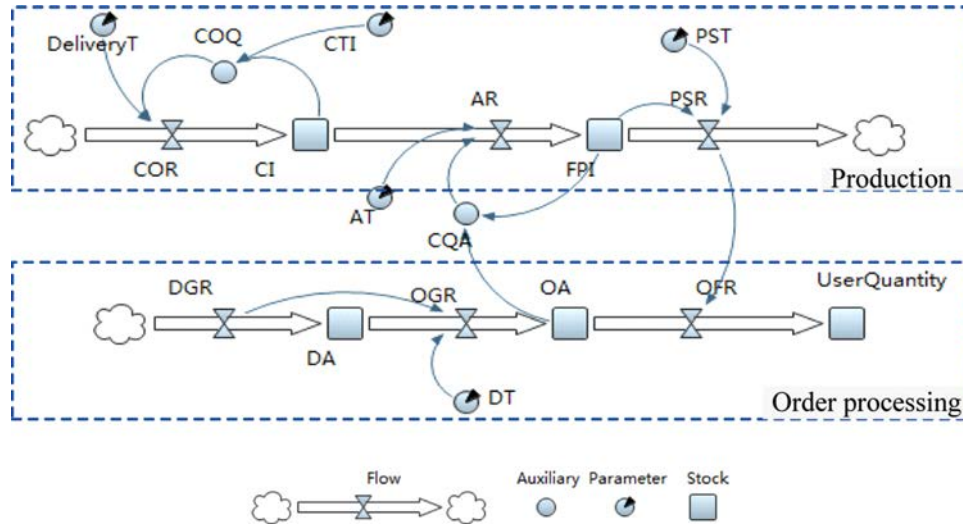


Fig. 6. User-customized configuration design pattern.

Table 1 Parameters settings for traditional design model.

Parameter	Initial value	Parameter	Initial value
Production Time (PT)	3	Processing Semi-Finished Products Inventory (PSFPI)	0
Delivery Time (DeliveryT)	3	Finished Products Inventory (FPI)	0
Manufacturing Time (MT)	3	Demand Accumulation (DA)	0
Product Shipping Time (PST)	3	Order Accumulation (OA)	0
Design Time (DT)	3	Demand Generate Rate (DGR)	Normal distribution (10,10)

Table 2 Flow settings for traditional design model.

Flow	Formula
Semi-Finished Products Order Rate (SFPOR)	$\frac{\text{Semi-Finished Products Order Quantity}}{\text{Preparation Time}}$
Manufacturing Rate (MR)	$\frac{\text{Processing Semi-Finished Products Inventory}}{\text{Manufacturing Time}}$
Product Shipping Rate (PSR)	$\frac{\text{Finished Products Inventory}}{\text{Product Shipping Time}}$
Preparation Time (PreT)	$\text{Production Time} + \text{Delivery Time}$
Semi-Finished Products Order Quantity (SFPOQ)	$\text{Order Accumulation} - \text{Processing Semi-Finished Products Inventory} - \text{Finished Products Inventory}$
Order Generate Rate From Demand (OGRFD)	$\frac{\text{Demand Accumulation}}{\text{Design Time}}$
Order Finished Rate (OFR)	$\text{Product Shipping Rate}$

Table 3 Parameters settings for user-customized configuration design model.

Parameter	Initial value	Parameter	Initial value
Delivery Time (DeliveryT)	3	Component Inventory (CI)	30
Component Target Inventory (CTI)	30	Finished Products Inventory (FPI)	0
Assembly Time (AT)	3	Demand Accumulation (DA)	0
Product Shipping Time (PST)	3	Order Accumulation (OA)	0
Design Time (DT)	3	Demand Generate Rate (DGR)	Normal distribution (10,10)

Table 4
Flow settings for user-customized configuration design model.

Flow	Formula
Component Order Rate (COR)	Component Order Quantity/Preparation Time
Assembly Rate (AR)	Component Quantity to be Assembled/Assembly Time
Product Shipping Rate (PSR)	Finished Products Inventory/Product Shipping Time
Component Order Quantity (COQ)	Components Target Inventory – Component Inventory
Order Generated Rate (OGR)	Delay (Demand Generated Rate, Design time)
Order Finished Rate (OFR)	Product Shipping Rate
Component Quantity to be Assembled (CQA)	Order Accumulation – Finished Products Inventory

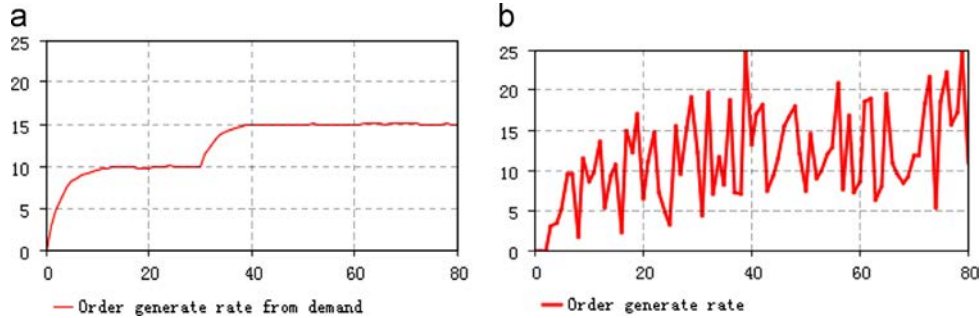


Fig. 7. (a) Order generated rate from demand in traditional design pattern and (b) order generated rate in user-customized configuration design pattern.

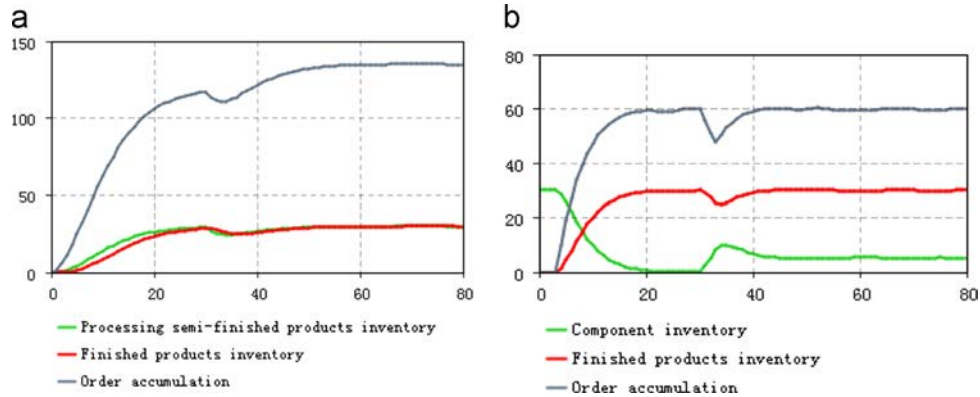


Fig. 8. (a) Simulation results of traditional design pattern and (b) simulation results of user-customized configuration design pattern.

- (2) When designing the same product, the performance of product designed through the two modes is the same.
- (3) In Fig. 4, the components in component library can fully meet customers' demand.

The companies organize production after receiving orders in these two patterns; thus order backlog will appear as shown in simulation results. In the case of the same user demand generated rate, order backlog of traditional pattern is more than 100, but the order backlog of the new pattern is only about 60. This result means that the response speed to user demand in user-customized design configuration pattern is much faster than in the traditional pattern. In addition, the inventory level directly determines the production cost of the products. From Fig. 8a and b, the total amount of processing semi-finished products and finished products inventory achieves stability at about 50 in traditional production pattern, while the total amount of component and finished products inventory achieves stability at about 30 in user-customized configuration design pattern. The result shows that the latter pattern ensures faster response speed to user needs in low inventory level.

When user demand mean is changed from 10 to 15, the inventories of these two patterns do not change significantly after

the system reaches a steady state. But in Fig. 8a, the order backlog increases about 10, and in Fig. 8b the order backlog does not change significantly. The result shows that the adaptability of the latter pattern is stronger when user demand is changed. In traditional design pattern, even if the company adjusts production capacity, the orders accumulation will still increase due to the transportation time constraints of semi-finished products.

In addition, the design time of the two patterns are set to 3, but Figs. 3 and 4 show that in traditional design pattern, the product is designed by the company, and in user-customized configuration design pattern, the product is designed by the user, which means there are as many designers as there are users. Therefore, from the perspective of the entire market, the actual design time of the new pattern is shorter than the traditional pattern, so as to shorten the time from demand generation to final product sent to users.

5. Conclusion

For the contradiction between high degree of customization and long lead time of industrial robot, the paper puts forward a

user-customized configuration design pattern, and some of the key issues relating to the pattern are studied, including

- (1) The implementation approach of this pattern is given through analyzing what role industrial robot stakeholders including suppliers, manufacturers, designers and users should play in the pattern.
- (2) Technology process of an open design platform is provided to manage the stakeholders in user-customized configuration design pattern implementation.
- (3) A preliminary solution of key issues for platform realization is given, including the use of I-Components to attract the companies developing components to provide enough components, using a customized template to provide technical support for users, providing component release criterion to support various stakeholders releasing and calling components.
- (4) System dynamics models were established for this design user-customized configuration pattern and for traditional design pattern of industrial robots by Anylogic simulation software. The simulation analysis proved that the proposed design pattern has better performances in terms of inventory cost and user order response quality than traditional design pattern.

In the initial set up stage, the types of industrial robots designed by users on the platform are very few, and some components needed to be redesigned by professional designers to meet most demands. But with the continuous development of the platform, as many more people are increasingly using the platform to design and purchase product, more and more components development companies or individual will publish their components on the platform for users calling to obtain benefits. When the number of components published to the component library is large enough, the majority of user demands can be met through the platform by user-customized configuration design. Then the advantages in terms of inventory cost and user order response quality will be reflected and the companies and users will achieve a win-win result.

Based on the proposed design pattern for user-customized configuration of industrial robot, further research work from platform implementation aspect is needed. Platform implementation technologies should integrate standardization of industrial robot components, development of customized template, building blocks of supply chain and qualified management of suppliers, components for quality ratings, etc.

Acknowledgment

This work was supported by graduate starting seed fund of Northwestern Polytechnical University (Grant no. Z2012171) and China Scholarship Council (CSC) (Grant no. 2011629137).

References

- [1] Fujita M, Kitano H, Kageyama K. A reconfigurable robot platform. *IEEE Robot Auton Syst* 1999;29(2-3):119–32.
- [2] Faiña A, Orjales F, Bellas F. First Steps towards a heterogeneous modular robotic architecture for intelligent industrial operation. In: *Proceedings of the workshop reconfigurable modular robotics: challenges of mechatronic and bio-chemo-hybrid systems, IROS*; 2011. p. 6.
- [3] Lemay J, Notash L. Configuration engine for architecture planning of modular parallel robots. *Mech Mach Theory* 2004;39:101–17.
- [4] Faiña A, Francisco B, Daniel S, Duro Richard J. Towards an evolutionary design of modular robots for industry. In: Ferrandez JM, Alvarez J.R., de la Paz F, Toledo F.J., (editors.), *IWINAC 2011, Part I, LNCS 6686*. pp. 50–9.
- [5] Mohammad Biglarbegian, Melek William W, Mendel Jerry M. Design of novel interval type-2 fuzzy controllers for modular and reconfigurable robots: theory and experiments. *IEEE Trans Ind Electron* 2011;58(4):1371–84.
- [6] Chen Chen-Yuan, Shih Bih-Yaw, Ma Jia-ming. Development for low-cost and cross-platform robot control environment. *J Sound Vib* 2013;29(1):228–33.
- [7] Plitea N, Lese D, Pislă D, Vaida C. Structural design and kinematics of a new parallel reconfigurable robot. *Robot Comput-Integr Manuf* 2013;29:219–35.
- [8] Baca J, Ferre M, Aracil R. A heterogeneous modular robotic design for fast response to a diversity of tasks. *Robot Auton Syst* 2012;60:522–31.
- [9] Lyder Andreas, Garcia Ricardo Franco Mendoza, Stoy Kasper. Mechanical design of Odin, an extendable heterogeneous deformable modular robot. In: *Proceedings of the IEEE/RSJ international conference on intelligent robots and systems*. Nice, France; September 22–26 2008.
- [10] Salvadora F, Forza C. Configuring products to address the customization responsiveness squeeze: a survey of management issues and opportunities. *Int J Prod Econ* 2004;91:273–91.
- [11] Pacheco M, Moghadam M, Magnusson A, Silverman B, Lund HH, Christensen DJ. Fable: design of a modular robotic playware platform. In: *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*. Karlsruhe, Germany; May 6–10 2013.
- [12] Bi ZM, Zhang WJ. Concurrent optimal design of modular robotic configuration. *J Robot Syst* 2001;18(2):77–87.
- [13] Miller Thomas D, Elgård Per. Defining modules, modularity and modularization-evolution of the concept in a historical perspective. design for integration in manufacturing. In: *Proceedings of the 13th IPS research seminar*. Fuglsoe; 1998.
- [14] Fukuda T, Nakagawa S. Dynamically reconfigurable robotic system. In: *Proceedings of the IEEE international robotics and automation conference*; 1988: vol. 3. p. 1581–6.
- [15] Wurst KH. The conception and construction of a modular robot system. In: *Proceedings of the International symposium on industrial robotics*. Belgium; 1986. p. 37–44.
- [16] Pavlic D, Storga M, Bojčević N, Marjanović D. Generic product structure of the configurable product. In: *Proceedings of the international design conference – Design 2004*. Dubrovnik, Croatia; May 18–21 2004.
- [17] Soininen T, Stumptner M. Configuration. *Artif Intell Eng Des. Anal Manuf* 2003;17(1):1–2.
- [18] Jiao J, Simpson TW, Siddique Z. Product family design and platform-based product development: a state-of-the-art review. *J Intell Manuf* 2007;18:5–29.
- [19] Tako AA, Robinson S. The application of discrete event simulation and system dynamics in the logistics and supply chain context. *Decis Support Syst* 2012;52:802–15.
- [20] Brunete A, Hernando M, Gambao E, Torres JE. A behavior-based control architecture for heterogeneous modular, multi-configurable, chained micro-robots. *Robot Auton Syst* 2012;60:1607–24.
- [21] Lau HYK, Ko AWY, Lau TL. The design of a representation and analysis method for modular self-reconfigurable robots. *Robot Comput-Integr Manuf* 2008;24: 258–69.
- [22] Eynard B, Gallet T, Roucoules L, Ducellier G. PDM System Implementation based on UML. *Math Comput Simul* 2006;70(5–6):247–442.
- [23] Olhager J. The role of the customer order decoupling point in production and supply chain management. *Comput Ind* 2010;61:863–8.
- [24] Guohua C, Genbao Z, Xuelin L. Study on dynamics behavior of supply chain with various operational modes. *Comput Integr Manuf Syst* 2011;17(1): 133–41.