

Article

A Time Correlation Based Clustering Method for a Design of a Transformable Product

Jinpu Zhang ^{1,2}, Guozhong Cao ^{1,2,*}, Qingjin Peng ³ , Runhua Tan ^{1,2}, Huangao Zhang ^{1,2} and Wei Liu ^{1,2}

¹ School of Mechanical Engineering, Hebei University of Technology, Tianjin 300401, China; jinpu_zhang@163.com (J.Z.); rhtanhebut@163.com (R.T.); zhgzwy@hebut.edu.cn (H.Z.); lwofhebut@126.com (W.L.)

² National Engineering Research Center for Technological Innovation Method and Tool, Tianjin 300401, China

³ Department of Mechanical Engineering, University of Manitoba, Winnipeg, MB R3T5V6, Canada; Qingjin.Peng@umanitoba.ca

* Correspondence: caoguzhong@hebut.edu.cn; Tel.: +86-132-1206-2906

Received: 6 December 2019; Accepted: 2 January 2020; Published: 5 January 2020



Abstract: A single-function product cannot meet various needs of different users when the product user or use environment changes. A transformable product with multiple functions can meet different needs of users. It is critical to determine product functions that a transformable product should have in the product design. However, it is a challenge to decide required functional components of a transformable product in the design process. A clustering method is proposed in this paper using undirected graphs for segmentations of needs in the time dimension. A need-based function model is built to form product function chains based on cost of the function transition distance between different function chains. Undirected graphs of the function chains are constructed according to the similarity of product functions. The interrelated subgraphs are then used to form multiple functions of a transformable product based on segmentations. A wheelchair is developed as an example to verify the proposed method. The method improves the design process of transformable products accurately and effectively.

Keywords: product design; transformable product; function chain; transition distance; graph representation

1. Introduction

A product has to meet diverse requirements of various users in order to be competitive in the market. Although different products have been developed to meet different user needs, many products provide only a single function, which cannot fully utilize capacity of product resources such as material and space for a variety of user needs. There are products with multiple functions in the market. Some of these products perform different functions through the structural reconfiguration, and some provide multiple functions using a fixed structure. Because of the importance of multi-functional products, different methods have been proposed for design of multi-functional products, such as the reconfigurable design, adaptive design, modular design, and product family design. There are two kinds of reconfigurable design applications: one is the reconfiguration of manufacturing system resources to meet different design requirements [1–4], and the other is the reconfiguration of product function modules to form different design solutions [5–7]. The adaptive design was proposed by Gu et al. [8] to design products that can meet different needs in product design and applications. Design adaptability refers to the ability of the existing design to create new or modified design to meet changeable requirements. Product adaptability refers to the ability of the product to meet

changing needs in applications [9]. The modular design is widely used to form a product by using independent function modules [10,11]. The module refers to a unit composed of a component or a group of components to perform a function of the product. Each module can be easily attached, detached, modified, relocated, and replaced for upgrading, modifying, recycling and reusing of the product [12,13]. Common modules of the product can be used as platforms [14,15]. The concept of the product platform can be used to design flexible products with similar common function modules to meet different customer needs [16,17].

These existing methods can effectively guide the design of multi-functional products. However, when there is a need for a product to change functions only through structural transformation without replacing components, the modular product will not be able to meet the requirement. A transformable product is necessary to meet the need. A product is called the transformable product that can change its physical structure to perform different functions or enhance existing functions. For example, the Osprey plane can be transformed between helicopter and aircraft, and Aquada can be formed as either a car or ship [18,19]. There are also many transformable products applied in the architecture and furniture industry [20–22]. Transformable products can often be smaller and lighter than equivalent single-state products or a set of single-function products [23]. Transformation is an action of changing the physical structure of a product in order to perform different functions or enhance an existing function. In order to effectively design transformable products, transformation design methods are required [24].

Comparing with a general product, the transformable product uses less material and space to meet different use requirements. It is critical to determine product functions that a transformable product should have in the product design. Although many approaches have been proposed for design of the transformable product, there is a lack of the effective way to identify all potential functions and select the appropriate set of functions for a transformable product. This paper reviews the related work of the design method of transformable products, and proposes a new method to decide required functional components of a transformable product in the design process. This paper focuses on the conceptual design stage of transformable products as the early stage of design determines 80% of product lifespan cost [25]. This paper proposes an effective approach to improve the design for transformable products. The approach integrates methods of needs clustering based on the time correlation, decision-making of the function chain similarity based on transition distances, and segmentation of graph representation.

We will first introduce the representation of users' needs using a standard form. The needs are clustered according to the time dimension. Function chains of the needs are then established after clustering, and the transition distance between function chains is determined according to the cost measure. The similarity between function chains is obtained according to the transition distance, and final functions of the transformable product are determined based on the similarity. A wheelchair is developed to verify our proposed method. The proposed method improves the conceptual design process of transformable products. A mathematical model is developed to make the design process of transformable products accurately and effectively. The systematic design process enables the computer-aided design of transformable products.

2. Related Research

Different methods have been proposed for design of transformable products. Singh et al. [24] developed transformation products through induction and deduction using three principles and twenty facilitators. A proposed transformation principle provides a general direction to form a certain type of mechanical transformation. The transformation facilitator is a design structure to help create mechanical transformation. Weaver et al. [23,26] analyzed the transformation process of a large number of transformable systems. The use frequency of transformation principles and facilitators in each system is counted. Correlations between transformation principles and facilitators are decided for their use priority in the design process. Transformation principles and facilitators can also be used together with modified mind-maps, and other methods such as T cards, word trees, case-based automated design, and example design [27–31]. Son et al. [32,33] studied the human behavior influenced by

transformable products to develop new tools for designers in conceptual design of transformable products. Difficulties that designers may encounter in designing transformable products were found, causes of the difficulties were analyzed, and the effective design process of transformable product was inferred [34]. A transformable product is normally designed based on following three considerations: (1) the product needs packaging for portability and deployment; (2) multiple functions are required in one system for the use convenience; (3) multiple-functions products with dissimilar configurations share common material and/or energy flow [27–29]. Four transformation indicators were proposed including sharing functions; adhering to a variable; accommodating a process; and storing [30]. The design solution can be improved to enhance quality, novelty, and feasibility by categorizing the problem context to decide whether a product is suitable for a transformable product or a single function product [31]. The above existing research involves many aspects of transformable products, but how to determine the final function of transformable products has not been solved.

As the transformable product is closely related to the need of users and use environments, a human–machine–environment system should be considered in the design of transformable products [35], as shown in Figure 1.

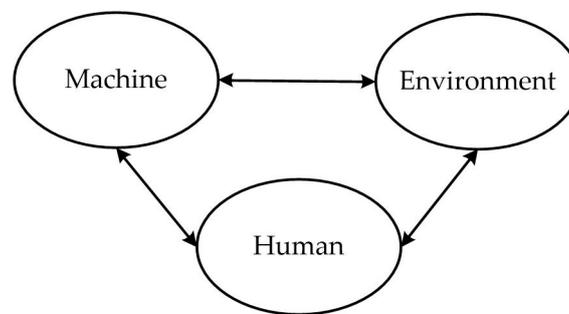


Figure 1. A human–machine–environment system [36].

For design of a product that can be changed to meet needs of environments or function changes, Desouza et al. [37] proposed a method to let users participate in the design process. Hazelrigg et al. [38] simplified the design into two steps: generating all possible design concepts and selecting the best solution. In order to fully obtain possible solutions, all the user needs have to be considered. Needs can be acquired through user interviews, group discussions, and Big Data analysis [39,40]. Needs are then processed by designers in product design to meet the needs. A normalized expression of needs was used to translate users' descriptions into design specifications [41]. An extension theory has been applied in product design to study possibility of function expanding using engineering laws and methods for innovation [42]. The extension theory can describe and analyze design using an extension matrix. Quality Function Deployment(QFD) is a multi-level deductive analysis method to transform customer requirements into design specifications, component characteristics, process requirements, and production requirements [43,44]. Axiomatic design divides design problems into four domains: user domain, function domain, structure domain, and process domain. User requirements, function requirements, design parameters, and process variables are applied in the design process. The mapping process from the user domain to function domain can guide the creativity of designers, the mapping process from a functional domain to physical domain can assist designers in conceptual design of transformable products [45–47]. Scenarios can be proposed for dynamic descriptions of user–system interactions [48]. Scenarios are usually narrated in the form of stories, including settings, actors, agent's goals, scenario's goals, plot, and timeframe [49]. Building scenarios based on needs can evolve functions of transformable products. Functional decomposition can recognize design problems. Through decomposition, designers can get a clear understanding of design problems and requirements. Many scholars have studied the process of functional decomposition from different perspectives [50–52]. A functional decomposition process is shown in Figure 2.

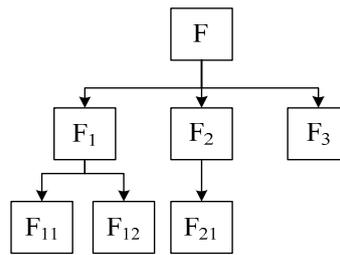


Figure 2. Functional decomposition [52].

The lowest sub-functions can be arranged in a logical relationship to form a function chain, which reflects the process of realizing the functions [53]. Analyzing the similarity between function chains of single function products can determine whether they should be merged into function chains of transformable products. There are two forms of the function chain: serial and parallel chains, as shown in Figure 3.

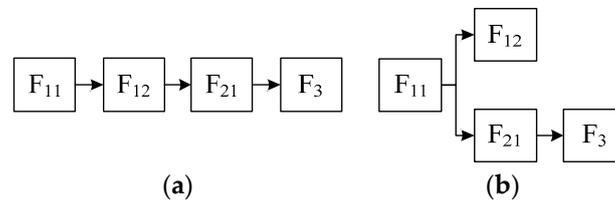


Figure 3. Two forms of the function chain [53]. (a) serial function chain; (b) parallel function chain.

Although these methods are not specially used to design transformable products, they can greatly assist the design for needs acquisition and function design of transformable products.

3. Proposed Method

For a large number of user needs, the need correlation can be analyzed using the extension theory as follows:

$$N = \begin{bmatrix} N & \text{User} & u_1 \\ & \text{Scene} & u_2 \\ & \text{Time} & u_3 \\ & \text{Activity} & u_4 \\ & \text{Current situation} & u_5 \\ & \text{Expected situation} & u_6 \end{bmatrix}.$$

User means people have the need. Scene is the scenario where the need exists. Time is when the need is required. Activity means actions to respond to the need. The current situation is the state of current activities at the time of the application. Expected situation means the expected activities at the time of the product application. Potential need sets of a basic product are represented by $N = [N_1, N_2, \dots, N_i, \dots, N_m]$. u_1-u_6 are specific descriptions of user, scene, time, activity, current situation, and expected situation. The importance of each need is determined by a comparing table [54] as shown in Table 1.

Table 1. Comparison of the importance of each need.

Needs	N ₁	N ₂	...	N _m
N ₁	N ₁	N ₂	...	N ₁
N ₂		N ₂	...	N ₂
...		
N _m				N _m

According to the frequency of needs appearing in the matrix, the importance of needs can be decided and recorded as w_i . For two needs with the high importance, the relationship between them will also be very important in the needs set. The sum of the two needs importance is used to express the weight of the two needs relationship. Dividing the sum by $m - 1$ is for the weight normalization, where m is the number of needs. The weight of the relationship between two needs is decided using Equation (1):

$$w_{N_i N_j} = \frac{w_{N_i} + w_{N_j}}{m - 1}. \tag{1}$$

Different functions of a transformable product are applied in different time periods [1], such as 0, 1, 3, 6, and 10, can be used to indicate the degree of time dependency [55]. Transformable products meet different needs at different times, so needs should be clustered in the time dimension to facilitate the design of transformable products. The time correlation between potential needs can be expressed in levels of 0, 0.3, 0.6, and 1. As shown in Equation (2),

$$\frac{\partial(C_{N_i}u_3)}{\partial(C_{N_j}u_3)} = \begin{cases} 0 & (C_{N_i}u_3) \cap (C_{N_j}u_3) = 0 \\ 0.3 & (C_{N_i}u_3) \cap (C_{N_j}u_3) \neq 0 \text{ and} \\ & (C_{N_i}u_3) \cup (C_{N_j}u_3) \neq [(C_{N_i}u_3) \text{ or } (C_{N_j}u_3)] \quad i, j \in [1, m]; \\ 0.6 & (C_{N_i}u_3) \cap (C_{N_j}u_3) = (C_{N_j}u_3) \text{ and} \quad i \neq j. \\ & (C_{N_i}u_3) \cup (C_{N_j}u_3) = (C_{N_i}u_3) \\ 1 & (C_{N_i}u_3) \cap (C_{N_j}u_3) = (C_{N_i}u_3) \cup (C_{N_j}u_3) \end{cases}, \tag{2}$$

where $(C_{N_i}u_3) \cap (C_{N_j}u_3) = 0$ represents that there is no intersection of two needs in the same time, $(C_{N_i}u_3) \cap (C_{N_j}u_3) \neq 0$ and $(C_{N_i}u_3) \cup (C_{N_j}u_3) \neq [(C_{N_i}u_3) \text{ or } (C_{N_j}u_3)]$ mean that there is a partial intersection of two needs in the use time.

$(C_{N_i}u_3) \cap (C_{N_j}u_3) = (C_{N_j}u_3)$ and $(C_{N_i}u_3) \cup (C_{N_j}u_3) = (C_{N_i}u_3)$ represent that the usage time of N_j is a true subset of the usage time N_i .

$(C_{N_i}u_3) \cap (C_{N_j}u_3) = (C_{N_i}u_3) \cup (C_{N_j}u_3)$ represents that N_i and N_j work at the same time.

The time related degree of all needs can be obtained by calculating the time correlation of each pair of two needs. According to Equations (1) and (2), the time independence between needs can be obtained using Equation (3):

$$r_{ij} = \frac{w_{N_i N_j}}{\text{Max}\{w_{N_i N_j}\}} \times \frac{\partial(C_{N_i}u_3)}{\partial(C_{N_j}u_3)}, \tag{3}$$

where $\text{Max}\{w_{N_i N_j}\}$ is the correction coefficient for values in the matrix easily distinguishable. According to Equation (3), the relationship matrix of needs can be established using Equation (4):

$$R = \begin{matrix} N_1 \\ N_2 \\ \vdots \\ N_m \end{matrix} \begin{bmatrix} r_{11} & r_{12} & & \\ & r_{22} & & \\ & & \ddots & \\ & & & r_{mm} \end{bmatrix}. \tag{4}$$

$N_1 \quad N_2 \quad \dots \quad N_m$

The graph theory uses a graph as its object of study to transform the relationship matrix into a graph [56]. According to the relationship matrix, we can draw a need-related undirected graph, in which nodes are needs and edges are the relationship between the needs. Segmentations of the undirected graph are decided by setting an appropriate threshold. Each interconnected subgraph forms a kind of need with the time dependence. Each kind of need corresponds to a potential state of the product. Function design is conducted for each product form to obtain a function chain of the product in each use state.

Function chains express sub-functions and their relationships. A directed graph can represent the logical relationship between nodes [56]. Function chains can be expressed by directed graphs, for example, $G = (F, L)$, where $F = \{F_1, F_2, \dots, F_n\}$, $L = \{L_1, L_2, \dots, L_n\}$. G is a function chain, F is a set of sub-functions of function chain G , and L is a set of relationships among sub-functions. Figure 4a shows a directed graph of a function chain in which a product performs one function. In practice, a product function can be used many times. A chain of multiple functions is shown in Figure 4b.

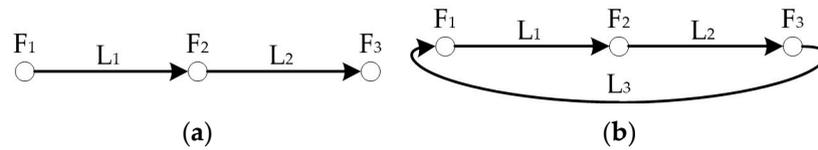


Figure 4. Function chains (a) function chain that performs one function; (b) function chain that performs multiple functions.

When a product needs to perform multiple functions, it needs to be transited from one function chain to another. Much research has been done on the transition from one graph to another [57–60]. The cost of a transition is directly related to its suitability. The lower the cost of the transition, the higher the possibility that the two function chains can be converted to each other. The higher the similarity between the two function chains, the easier conversion of the chains. The transition from one function chain to another requires a series of operations, including the node converting, deleting, adding, and edge deletion and addition. Two function chains G_1 and G_2 are shown in Figure 5.

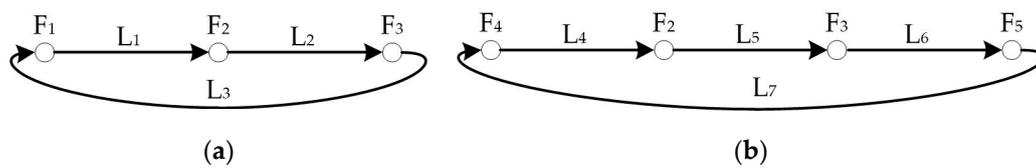


Figure 5. Two function chains (a) function chain G_1 ; (b) function chain G_2 .

The sum of costs of all operations from G_1 to G_2 is called the distance of transition from G_1 to G_2 , expressed by $Trans\{G_1, G_2\}$. The similarity between G_1 and G_2 is expressed as $Sim\{G_1, G_2\}$ using Equation (5):

$$Sim\{G_1, G_2\} = 1 - \frac{Trans\{G_1, G_2\}}{\sum |G_i|}, \tag{5}$$

where $|G_i|$ is the total number of nodes and edges in G_1 and G_2 . The distance of transition from G_1 to G_2 is calculated by Equation (6):

$$Trans\{G_1, G_2\} = con(F_1, F_4) + add(F_5) + add(L_6) + add(L_7) + add(L_3), \tag{6}$$

where $con(F_i, F_j)$ converts node F_i to node F_j , $del(F_i)$ deletes node F_i , $add(F_i)$ adds node F_i , $del(L_i)$ deletes edge L_i , and $add(L_i)$ adds edge L_i .

The cost of deleting and adding one node or edge can be assigned as one unit currency. Because there may be some similarity between two nodes, the cost of converting nodes is designed as $1 - Sim\{F_i, F_j\}$. $Sim\{F_i, F_j\}$ is the similarity between nodes F_i and F_j . The higher the similarity, the lower the cost of conversion. Semantic similarity computation has many applications in information retrieval, information extraction, word sense disambiguation, and machine translation [61,62]. In this paper, we use it to calculate similarity between two nodes based on the semantic similarity. All words are stored on a word tree according to the semantic relationship. The similarity of words can be calculated based on their distance in the word tree. The similarity of nodes is calculated by numbering them for their distance on the word tree using Equations (7)–(10) [63].

(1) If two words are not in the same tree.

$$Sim\{F_i, F_j\} = 0.1. \tag{7}$$

(2) If two words are in the same tree.

$$Sim\{F_i, F_j\} = \cos\left(n \times \frac{\pi}{180}\right) \left(\frac{n-k+1}{n}\right) \times \begin{cases} 0.65 & \text{In the second branch;} \\ 0.80 & \text{In the third branch;} \\ 0.90 & \text{In the fourth branch;} \\ 0.96 & \text{In the fifth branch;} \end{cases} \tag{8}$$

where n is the total number of nodes in the branch layer, and k is the distance between two branches.

(3) When numbers are same and the end number is '='.

$$Sim\{F_i, F_j\} = 1, \tag{9}$$

(4) When numbers are same and the end number is '#'.

$$Sim\{F_i, F_j\} = 0.5. \tag{10}$$

Using the semantic similarity calculation software of the synonym forest, we can calculate the similarity of two words according to the above definitions [64]. After calculating transition distances of function chains, the transition distances are weighted according to Equation (11):

$$w_{S_p S_q} = \frac{\sum w_{N_i} + \sum w_{N_j}}{z - 1} \quad N_i \in S_p \quad N_j \in S_q, \tag{11}$$

where z is the number of function chains, divided by $z - 1$ to make the sum of the weights be 1. The weighted transition distance of two function chains is calculated by Equation (12):

$$WSim\{G_p, G_q\} = \frac{w_{S_p S_q} Sim\{G_p, G_q\}}{Max\{w_{S_p S_q}\}}, \tag{12}$$

where $Max\{w_{S_p S_q}\}$ is the correction coefficient to make values in the matrix easily distinguishable.

For an undirected graph $A = (G, W)$, where $G = \{G_1, G_2, \dots, G_n\}$, $W = \{W_1, W_2, \dots, W_m\}$. G is a set of function chains, and W is a set of similarity between function chains. Using threshold value W_0 removes edges less than W_0 in the undirected graph and deletes independent nodes. A new subgraph A_0 is generated. Interconnected nodes in subgraph A_0 are the function chains of the transformable product. The connection between nodes indicates that the function chains represented by these nodes can be transformed each other. Each interconnected node represents a state of a transformable product, and these nodes form the concept of a transformable product. As it is difficult to map the functional domain to physical domain directly, physical components can be designed by refining functions in the mapping process, as shown in Figure 6.

For a certain function, the solution to realize the function can be selected by designers firstly. Components for implementing the function are then determined. The number of identical components in each state represents the space and material savings of the transformable product. The effectiveness of the method can be evaluated according to the proportion of shared components, as shown in Equation (13):

$$P = \frac{NCT}{\sum NCS_i}, \tag{13}$$

where P is the proportion of shared components, NCT is the number of components in the transformable product, and NCS_i is the number of components in the i -th state. The trend of technological evolutions

is used to improve the level of idealization [65]. The calculation equation of idealization levels is as follows:

$$I = \frac{\sum UF}{\sum HF + \sum EX'} \tag{14}$$

where I is the idealized level, $\sum UF$ is the sum of useful functions, $\sum HF$ is the sum of harmful functions, and $\sum EX$ is expenses including cost, space, consumed energy, noise, etc. In this paper, $\sum UF$ is the number of useful functions, $\sum HF$ is assumed as 0 because the harmful function should not be proposed in design, and $\sum EX$ is the number of components in the product.

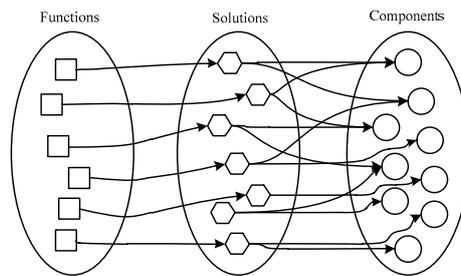


Figure 6. Mapping of functions–solutions–components.

Comparing the idealization level of a transformable product to multiple products with the same functions can effectively determine whether the function implementation has been improved. The idealization level of a transformable product divided by the idealization level of existing products is the change degree of the idealization level, as shown in Equation (15):

$$C = \frac{I_t}{I_e} \tag{15}$$

where C is the change degree of the idealization level, I_t is the idealization level of a transformable product, and I_e is the idealization level of existing products.

4. Case Study

A wheelchair is developed in a case study to verify our proposed method. The wheelchair is required to meet needs of people with walking difficulty due to illness, injury, or disability [66]. Different user groups or different usage environments may have different needs for the wheelchair. The collected user needs are listed in Table 2.

Table 2. User needs for a wheelchair.

No.	Needs
1	Climbing stairs without the other help
2	Backrest can be adjusted
3	Avoiding sun and rain in wheelchair
4	Sleep in wheelchair
5	Working in wheelchair
6	Moving for navigation
7	Using in the bathroom
8	Taking a bath in a wheelchair

These needs are formalized for details as shown in Table 3.

Table 3. User needs.

No.	User	Time	Scene	Activity	Current Situation	Expected Situation
N ₁	All the user	Travel time	Stair case	Climb the stairs	With other’s help	Without help
N ₂	Paralyzed user	Changing posture	Indoor	Sitting	Seat angle not adjustable	Seat angle adjustable
N ₃	All the user	Where there is sunshine or rain	Outdoor	Driving wheelchair forward	No shelter	Have shelter
N ₄	All the user	Afternoon	Indoor	Sleep	Sit down	Lie down
N ₅	Business people	Work time	Office	Work	Work at desk	Work in wheelchair
N ₆	Outgoing user	When going out	Outdoor	Driving wheelchair forward	No navigation	Have navigation
N ₇	Paralyzed user	When going to the toilet	Bathroom	Use the toilet	With help	Without help
N ₈	Paralyzed user	When taking a bath	Bathroom	Take a bath	With help	Without help

Needs in Table 2 are compared for the importance as shown in Table 4.

Table 4. Comparisons for the importance of needs for a wheelchair.

Needs	N ₁	N ₂	N ₃	N ₄	N ₅	N ₆	N ₇	N ₈
N ₁	N ₁	N ₂	N ₁	N ₁	N ₁	N ₁	N ₇	N ₈
N ₂		N ₂						
N ₃			N ₃	N ₄	N ₃	N ₃	N ₇	N ₈
N ₄				N ₄	N ₄	N ₄	N ₇	N ₈
N ₅					N ₅	N ₅	N ₇	N ₈
N ₆						N ₆	N ₇	N ₈
N ₇							N ₇	N ₇
N ₈								N ₈

The needs are weighted based on results of Table 4, as shown in Table 5.

Table 5. Weight of needs.

Needs N	Weight W
N ₁	0.14
N ₂	0.22
N ₃	0.08
N ₄	0.11
N ₅	0.06
N ₆	0.03
N ₇	0.19
N ₈	0.17

Weights of needs in Table 5 were decided by using the method proposed in Table 1. All needs are compared by a pairwise method. For two needs, the more important one is listed in Table 4 after the comparison. The final weight of a need is decided by its appearance times in Table 4. Weights in

Table 5 are normalized values; therefore, the total weights are 1. According to Equation (1), $w_{N_iN_j}$ is calculated as shown in Equation (16):

$$w_{N_iN_j} = \begin{matrix} N_1 \\ N_2 \\ N_3 \\ N_4 \\ N_5 \\ N_6 \\ N_7 \\ N_8 \end{matrix} \begin{bmatrix} 0 & 0.051 & 0.031 & 0.036 & 0.028 & 0.024 & 0.047 & 0.044 \\ & 0 & 0.043 & 0.047 & 0.04 & 0.036 & 0.059 & 0.055 \\ & & 0 & 0.027 & 0.02 & 0.016 & 0.039 & 0.038 \\ & & & 0 & 0.024 & 0.02 & 0.043 & 0.04 \\ & & & & 0 & 0.013 & 0.036 & 0.033 \\ & & & & & 0 & 0.031 & 0.028 \\ & & & & & & 0 & 0.051 \\ & & & & & & & 0 \end{bmatrix} \quad (16)$$

Using Equation (3), the relationship matrix of needs is established as shown in Equation (17):

$$R = \begin{matrix} N_1 \\ N_2 \\ N_3 \\ N_4 \\ N_5 \\ N_6 \\ N_7 \\ N_8 \end{matrix} \begin{bmatrix} 0 & 0.26 & 0.16 & 0 & 0 & 0 & 0 & 0 \\ & 0 & 0.22 & 0.48 & 0.41 & 0.18 & 0.3 & 0.28 \\ & & 0 & 0.14 & 0 & 0.08 & 0 & 0 \\ & & & 0 & 0 & 0 & 0 & 0 \\ & & & & 0 & 0 & 0 & 0 \\ & & & & & 0 & 0 & 0 \\ & & & & & & 0 & 0 \\ & & & & & & & 0 \end{bmatrix} \quad (17)$$

According to the relationship matrix, we can draw the needs-related undirected graph as shown in Figure 7.

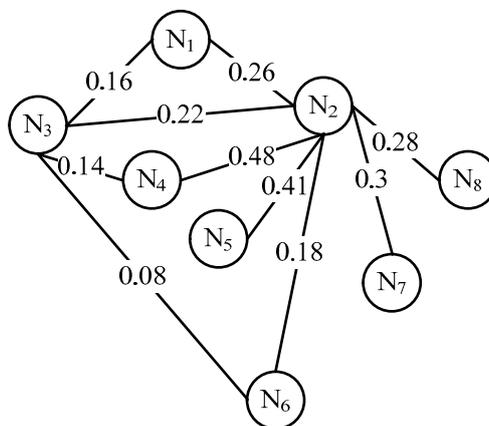


Figure 7. Needs-related undirected graph.

Designers can choose different thresholds in different scenes based on the design requirements. For a given threshold value $R = 0.4$, the resulting segmentations of the graph are shown in Figure 8.

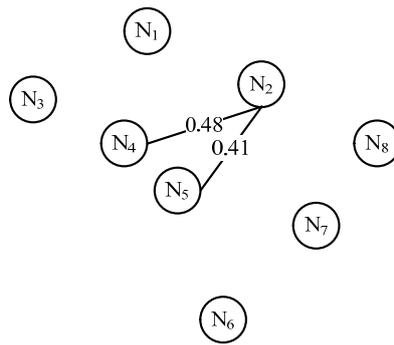


Figure 8. Segmentations of needs-related undirected graph.

As shown in Figure 8, the needs are divided into six categories corresponding to six potential states, as shown in Table 6.

Table 6. Scenario description of six states.

States	Needs	Scenario Description
S ₁	N ₁	Go downstairs in a wheelchair, climb stairs to a smooth road and continue to drive the wheelchair forward.
S ₂	N ₂ , N ₄ , N ₅	Adjust angle of the seatback to change sitting posture, adjust table board for office use, and adjust seat to sleep flat during break.
S ₃	N ₃	Drive the wheelchair in rainy days, adjust the shelter to block wind and rain.
S ₄	N ₆	Guide road when driving a wheelchair
S ₅	N ₇	Go to toilet and clean in wheelchair
S ₆	N ₈	Take a bath in wheelchair

The product functions in the scenario are obtained through the description of the scenario. Function chains of the six states are obtained after the function decomposition as shown in Figure 9.

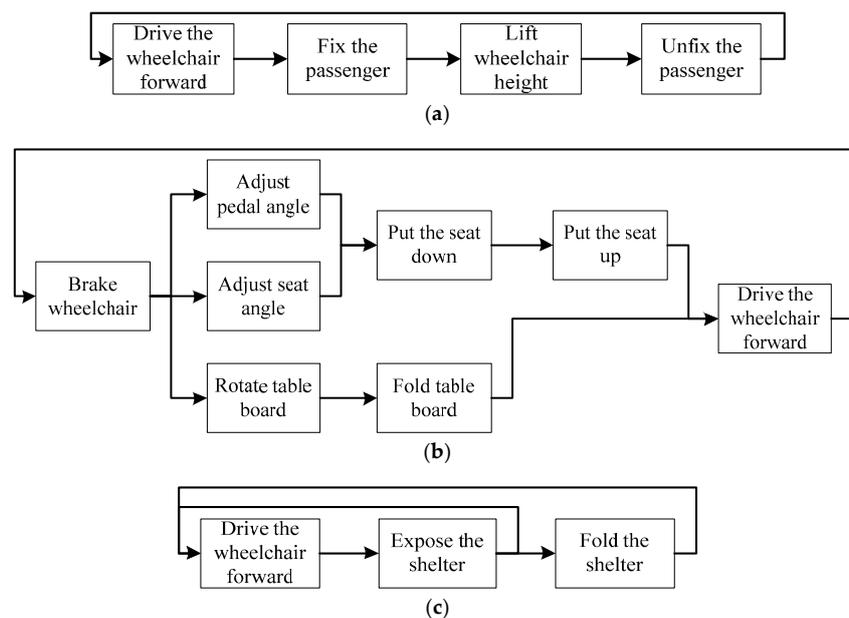


Figure 9. Cont.

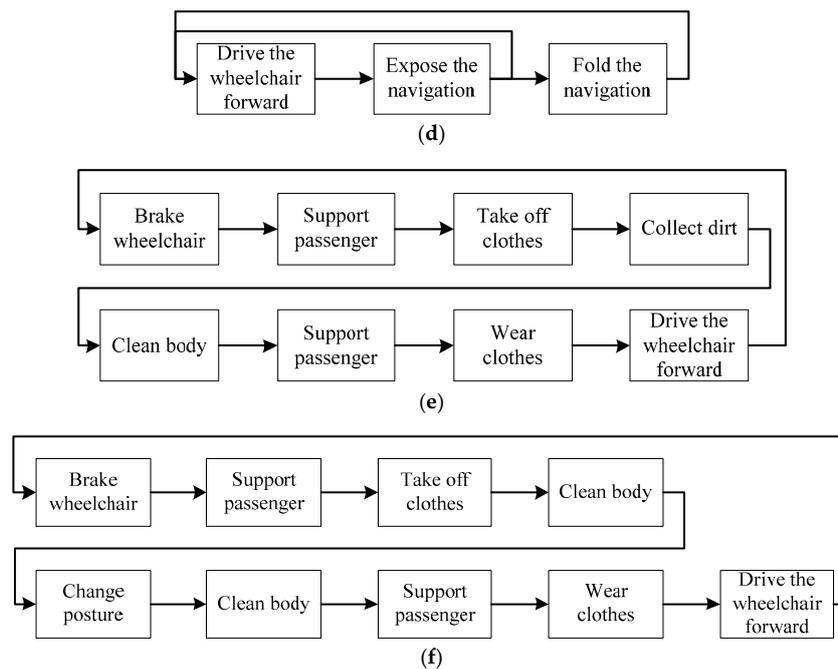


Figure 9. Function chains of the six states. (a) function chain G_1 , (b) function chain G_2 , (c) function chain G_3 , (d) function chain G_4 , (e) function chain G_5 , (f) function chain G_6 .

According to Equations (7)–(10), transition distances between any two function chains can be obtained as shown in Equation (18):

$$Trans\{G_p, G_q\} = \begin{matrix} S_1 \\ S_2 \\ S_3 \\ S_4 \\ S_5 \\ S_6 \end{matrix} \begin{bmatrix} 0 & 12.48 & 5.67 & 5.8 & 19.49 & 21.49 \\ & 0 & 20.8 & 20.35 & 28 & 30 \\ & & 0 & 0.9 & 17 & 19 \\ & & & 0 & 17 & 19 \\ & & & & 0 & 4.9 \\ & & & & & 0 \end{bmatrix} \begin{matrix} S_1 \\ S_2 \\ S_3 \\ S_4 \\ S_5 \\ S_6 \end{matrix} \quad (18)$$

Using Equation (5), the similarity of two functional chains is obtained as shown in Equation (19):

$$Sim\{G_p, G_q\} = \begin{matrix} S_1 \\ S_2 \\ S_3 \\ S_4 \\ S_5 \\ S_6 \end{matrix} \begin{bmatrix} 0 & 0.52 & 0.62 & 0.61 & 0.19 & 0.17 \\ & 0 & 0.17 & 0.19 & 0.18 & 0.17 \\ & & 0 & 0.94 & 0.26 & 0.24 \\ & & & 0 & 0.26 & 0.24 \\ & & & & 0 & 0.86 \\ & & & & & 0 \end{bmatrix} \begin{matrix} S_1 \\ S_2 \\ S_3 \\ S_4 \\ S_5 \\ S_6 \end{matrix} \quad (19)$$

Based on Equation (11), weights of the distance between any two function chains are obtained as shown in Equation (20):

$$\omega_{S_i S_j} = \begin{matrix} S_1 \\ S_2 \\ S_3 \\ S_4 \\ S_5 \\ S_6 \end{matrix} \begin{bmatrix} 0 & 0.106 & 0.044 & 0.034 & 0.066 & 0.062 \\ & 0 & 0.094 & 0.084 & 0.116 & 0.112 \\ & & 0 & 0.022 & 0.054 & 0.050 \\ & & & 0 & 0.044 & 0.040 \\ & & & & 0 & 0.072 \\ & & & & & 0 \end{bmatrix} \begin{matrix} S_1 \\ S_2 \\ S_3 \\ S_4 \\ S_5 \\ S_6 \end{matrix} \quad (20)$$

According to Equation (12), the weighted similarity is obtained as shown in Equation (21):

$$WSim\{G_p, G_q\} = \begin{matrix} S_1 \\ S_2 \\ S_3 \\ S_4 \\ S_5 \\ S_6 \end{matrix} \begin{bmatrix} 0 & 0.48 & 0.24 & 0.47 & 0.11 & 0.09 \\ & 0 & 0.14 & 0.14 & 0.18 & 0.16 \\ & & 0 & 0.18 & 0.12 & 0.10 \\ & & & 0 & 0.10 & 0.08 \\ & & & & 0 & 0.53 \\ & & & & & 0 \end{bmatrix} \begin{matrix} S_1 \\ S_2 \\ S_3 \\ S_4 \\ S_5 \\ S_6 \end{matrix} \quad (21)$$

An undirected graph of the matrix in Equation (21) is then established as shown in Figure 10.

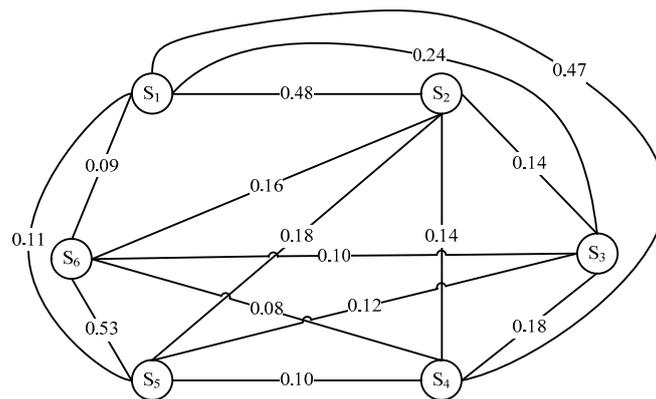


Figure 10. The undirected graph of the weighted similarity.

For a given threshold value $R = 0.4$, the resulting segmentation of the graph is formed in Figure 11.

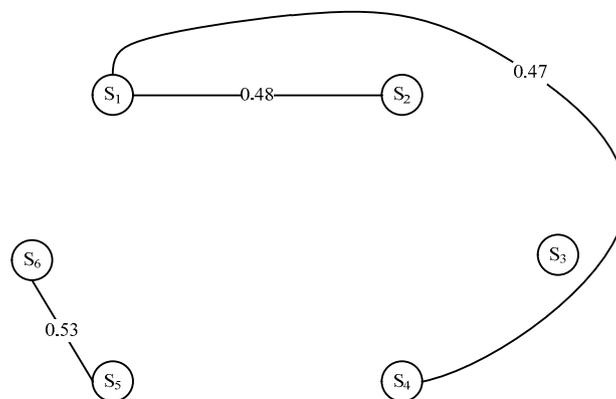


Figure 11. Segmentations of the undirected graph based on the weighted similarity.

Therefore, two final concepts can be formed for the wheelchair based on two segments formed in Figure 11, and the first concept has three states S_1, S_2 , and S_4 . Its function chain is shown in Figure 12.

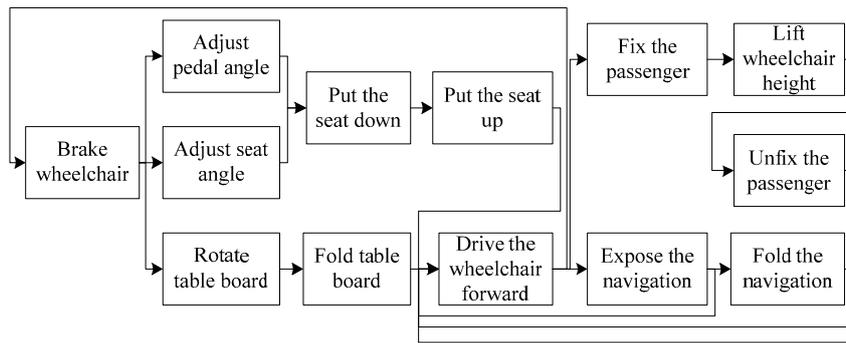


Figure 12. Function chain of the first concept.

Following the mapping process in Figure 6, the relationship between functions and components of the first concept can be formed as shown in Table 7. Based on corresponding components to meet the required functions, a conceptual model of the wheelchair can be proposed. The computer-aided design(CAD) model of the first concept is shown in Figure 13.

Table 7. Relationship between functions and components of the first concept.

States	Functions	Components
Climbing state	Drive the wheelchair forward	Wheels
	Fix the passenger	Wheel rack
	Lift wheelchair height	Frame
	Unfix the passenger	Seat plate
	Brake wheelchair	Backrest
Working and sleeping state	Adjust pedal angle	Handrail
	Adjust seat angle	Track
	Rotate table board	Track frame
	Put the seat down	Track wheel
	Fold table board	Pedal hinge
	Put the seat up	Pedal
	Drive the wheelchair forward	Backrest hinge
Navigation state	Drive the wheelchair forward	Board
	Expose the navigation	Navigation rack
	Fold the navigation	Navigation rack

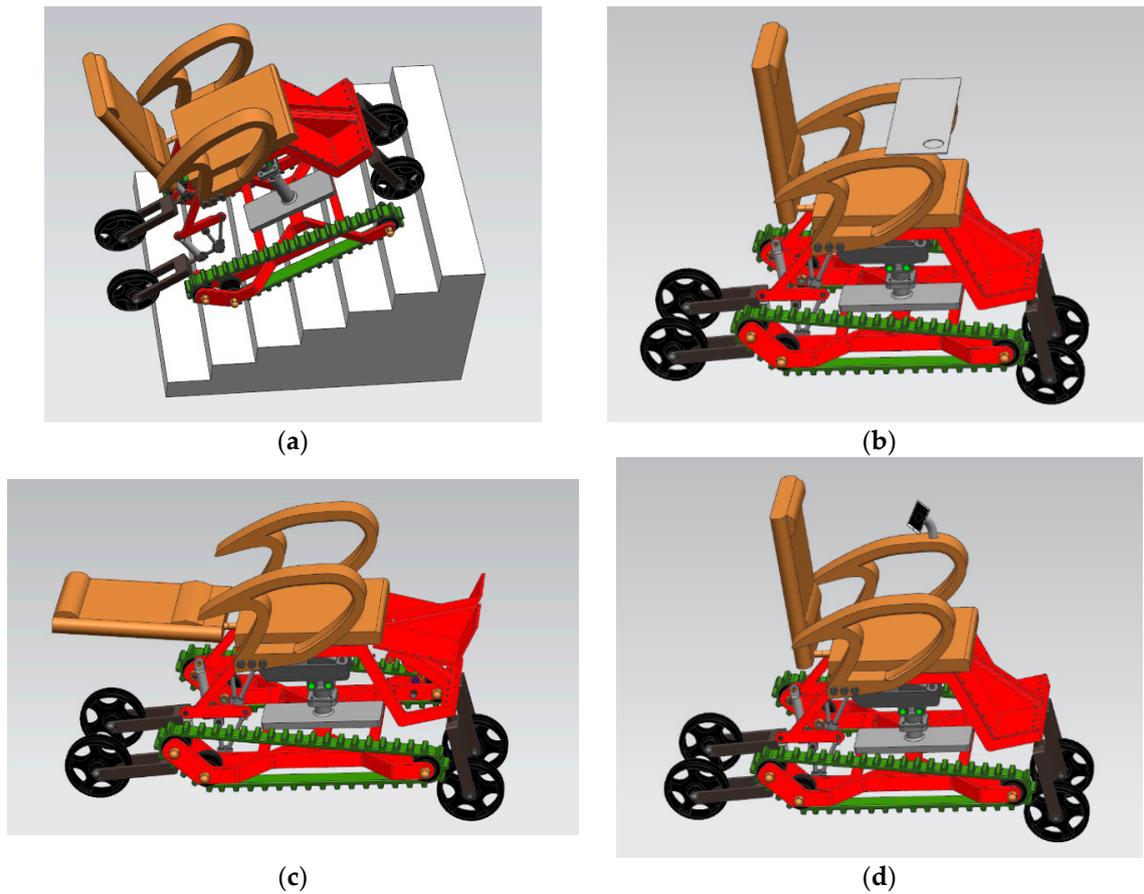


Figure 13. The first conceptual design of the wheelchair. (a) in climbing state, (b) in working state, (c) in sleeping state, (d) in navigation state.

The second concept consists of S_5 and S_6 , its function chain is shown in Figure 14.

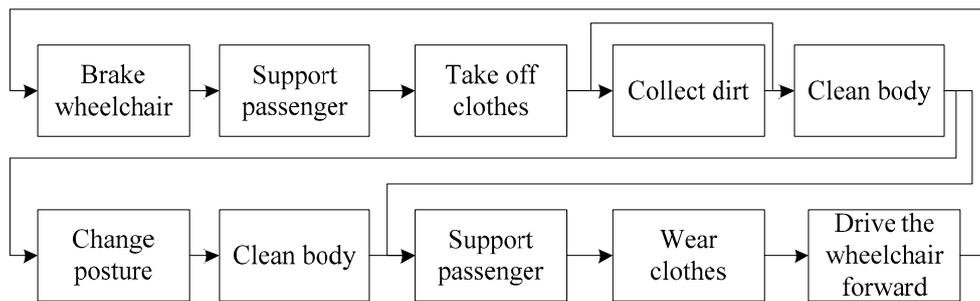


Figure 14. Function chain of the second design concept.

According to the mapping relationship of functions–solutions–components in Figure 6, the relationship between functions and components of the second concept is mapped as shown in Table 8.

Table 8. Relationship between functions and components of the second concept.



The CAD model of the second conceptual design is shown in Figure 15.

According to Equation (13), the proportion of shared components in the first concept can be calculated as 70%. The proportion of shared components in the second concept can be calculated as 75%. More than half of the parts are reused, which saves a lot of material and space resources compared to the single-function wheelchairs. The number of useful functions and number of components are shown in Table 9.

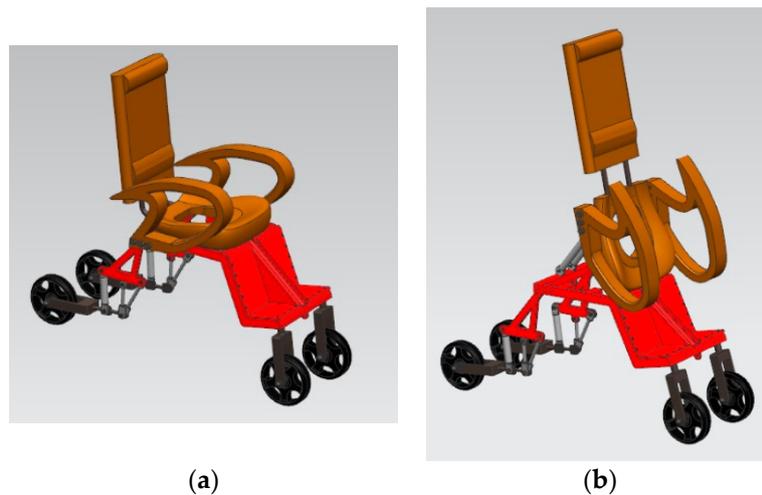


Figure 15. The second design of the wheelchair. (a) in toilet state, (b) in bathing state.

Table 9. Useful functions and related components in two concepts.

Product	Number of Useful Functions	Number of Components
Transformable product in first concept	13	14
Multiple single-function products in first concept	13	20
Transformable product in second concept	8	9
Multiple single-function products in second concept	8	12

According to Equations (14) and (15), the change degree of the idealization level can be calculated as 1.43 in the first concept, and 1.33 in the second concept. The results show that our proposed method increases the level of the product idealization and conforms to the trend of technological system evolutions in the design for transformable products.

5. Discussion

A transformable product uses the less resource and space than single-function products. Methods were proposed in this paper to decide required functional components and conceptual design of transformable products. Although there are many methods proposed to decide the need and design of transformable products [27–31], the method proposed in this research improves the existing methods by providing an effective tool to assist designers in the design of transformable products. A mathematical model was established for the design process of transformable products accurately and effectively. It proposed the time correlation to cluster the needs, which is a quantitative clustering method that can better assist designers to cluster needs. Segmentations of needs were used to form scenarios of function chains. The transition distance was determined by the cost of transition between different function chains. By calculating the distance between function chains, the similarity between them are more objectively. A weighted undirected graph was built according to the transition distance. After the undirected graph segmentation, the connected subgraphs were used to form final design concepts of transformable products. The segmentation of graph makes the final concept more systematic. The case study verified our proposed method in the design of a transformable wheelchair. Compared to the existing wheelchairs on the market that can only drive passengers on the smooth road, two concepts of the proposed transformable wheelchair can meet various needs of operation functions. Although the number of the possible detailed constructions of wheelchairs is huge or infinite, the conceptual design of wheelchairs to meet transformable requirements is limited considering the available functional components and configurations. We used benchmarking and expert evaluations to generate concepts and select the feasible solution. The design results were evaluated by comparing the idealization level

of transformable products with the existing products. The two concepts generated in the case study demonstrated effectiveness of the proposed method. The results show that our proposed method increases the level of the wheelchair idealization and conforms to the trend of technological system evolutions in the design for transformable products. In this paper, we evaluate results by comparing the idealization level of transformable products with the existing products. If the idealization level is improved by the proposed method, it shows that the method is effective. We are working on considering more factors to evaluate transformable products. The optional solution will be searched in the further work. It will consider not only the technical implementation but also expectations of the industry and users to evaluate transformable products. The method improves the conceptual design process of transformable products. However, the process is mainly a manual process which needs the design knowledge of designers. The current method is for the redesign of an existing product, not for development of new products. The design process is mainly a manual process that needs the design knowledge of designers. The further work will extend the method to the new product design. More factors will be considered to search an optional solution and evaluate design solutions. The method will include not only the technical implementation but also expectations of the industry and users. A software tool will be developed for the automatic design of transformable products. Further work will also apply the proposed method to different transformable products for the method improvement. Interfaces of transformable products will be considered. A machine learning method will be used to decide the change need of transformable products from one state to another.

Author Contributions: Conceptualization, J.Z., G.C., and H.Z.; methodology, J.Z., G.C., Q.P., and R.T.; writing—original draft preparation, J.Z.; writing—review and editing, Q.P., W.L., and J.Z.; supervision, G.C.; project administration, R.T.; funding acquisition, G.C. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the National innovation method work special project, Grant No. 2019IM020200, the Innovation Ability Promotion Project of Hebei Province, Grant No. 18961823H, the Support Scheme for 100 Outstanding Innovative Talents in Colleges and Universities in Hebei Province, Grant No. SLRC2017030, the Natural Science Foundation of China, Grant No. 51675159, the Central Government Guides Local Science and Technology Development Project, Grant No. 18241837G, and the Project Supported by Hebei Provincial Department of Education for Innovation Ability Training of Graduate Students, Grant No. cxzbs2019033.

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

References

1. Koren, Y.; Heisel, U.; Jovane, F.; Moriwaki, T.; Pritschow, G.; Ulsoy, G.; Van Brussel, H. Reconfigurable manufacturing systems. *CIRP Ann.* **1999**, *48*, 527–540. [[CrossRef](#)]
2. ElMaraghy, H.A. *Changeable and Reconfigurable Manufacturing Systems*; Springer: London, UK, 2008.
3. Musharavati, F. A method for optimizing the reconfiguration design task in manufacturing systems. In Proceedings of the 2011 International Conference on Industrial Engineering and Operations Management, Kuala Lumpur, Malaysia, 22–24 January 2011; pp. 603–609.
4. Aguilar, A.; Roman-Flores, A.; Huegel, J.C. Design, refinement, implementation and prototype testing of a reconfigurable lathe-mill. *J. Manuf. Syst.* **2013**, *32*, 364–371. [[CrossRef](#)]
5. Carbonari, L.; Callegari, M.; Palmieri, G.; Palpacelli, M.-C. A new class of reconfigurable parallel kinematic machines. *Mech. Mach. Theory* **2014**, *79*, 173–183. [[CrossRef](#)]
6. Katz, R. Design principles of reconfigurable machines. *Int. J. Adv. Manuf. Technol.* **2007**, *34*, 430–439. [[CrossRef](#)]
7. Bi, Z.M.; Lang, S.Y.T.; Verner, M.; Orban, P. Development of reconfigurable machines. *Int. J. Adv. Manuf. Technol.* **2008**, *39*, 1227–1251. [[CrossRef](#)]
8. Gu, P.; Hashemian, M.; Nee, A.Y.C. Adaptable design. *CIRP Ann. Manuf. Technol.* **2004**, *53*, 539–557. [[CrossRef](#)]
9. Gu, P.; Xue, D.; Nee, A.Y.C. Adaptable design: Concepts, methods, and applications. *Proc. Inst. Mech. Eng. Part B J. Eng. Manuf.* **2009**, *223*, 1367–1387. [[CrossRef](#)]
10. Gershenson, J.K.; Prasad, G.J.; Zhang, Y. Product modularity: Definitions and benefits. *J. Eng. Des.* **2003**, *14*, 295–313. [[CrossRef](#)]

11. Gershenson, J.K.; Prasad, G.J.; Zhang, Y. Product modularity: Measures and design methods. *J. Eng. Des.* **2004**, *15*, 33–51. [[CrossRef](#)]
12. Jung, S.; Simpson, T.W. New modularity indices for modularity assessment and clustering of product architecture. *J. Eng. Des.* **2017**, *28*, 1–22. [[CrossRef](#)]
13. Kashkoush, M.; Elmaraghy, H. Designing modular product architecture for optimal overall product modularity. *J. Eng. Des.* **2017**, *28*, 293–316. [[CrossRef](#)]
14. Simpson, T.W.; Siddique, Z.; Jiao, J. *Product Platform and Product Family Design, Methods and Applications*; Springer: Berlin, Germany, 2006.
15. Jiao, J.; Simpson, T.W.; Siddique, Z. Product family design and platform-based product development: A state-of-the-art review. *J. Intell. Manuf.* **2007**, *18*, 5–29. [[CrossRef](#)]
16. Saleh, J.H.; Mark, G.; Jordan, N.C. Flexibility: A multi-disciplinary literature review and a research agenda for designing flexible engineering systems. *J. Eng. Des.* **2009**, *20*, 307–323. [[CrossRef](#)]
17. Palani-Rajan, P.K.; Van-Wie, M.; Campbell, M.I.; Wood, K.L.; Otto, K.N. An empirical foundation for product flexibility. *Des. Stud.* **2005**, *26*, 405–438. [[CrossRef](#)]
18. BELL-BOEING V-22. Available online: <https://www.skybrary.aero/index.php/V22> (accessed on 5 December 2019).
19. Aquada. Available online: <https://www.gibbsamphibians.com/platform/aquada/> (accessed on 5 December 2019).
20. Liapi, K. Transformable structures: Design features and preliminary investigation. *J. Archit. Eng.* **2001**, *7*, 13–17. [[CrossRef](#)]
21. Andjelkovic, V. Transformation principles in the architectural design of a contemporary house. *Arch. Doct* **2016**, *4*, 87–107.
22. HOLMSUND. Available online: <https://www.ikea.cn/cn/en/catalog/products/S09228206/> (accessed on 5 December 2019).
23. Weaver, J.; Wood, K.; Crawford, R.; Jensen, D. Transformation design theory: A meta-analogical framework. *J. Comput. Inf. Sci. Eng.* **2010**, *10*, 031012. [[CrossRef](#)]
24. Singh, V.; Skiles, S.M.; Krager, J.E.; Wood, K.L.; Jensen, D.; Sierakowski, R. Innovations in design through transformation: A fundamental study of transformation principles. *J. Mech. Des.* **2009**, *131*, 081010. [[CrossRef](#)]
25. Dowlatshahi, S. Product design in a concurrent engineering environment: An optimization approach. *Int. J. Prod. Res.* **1992**, *30*, 1803–1818. [[CrossRef](#)]
26. Weaver, J. Transformer Design: Empirical Studies of Transformation Principles, Facilitators, and Functions. Master's Thesis, University of Texas at Austin, Austin, TX, USA, 2008.
27. Singh, V.; Walther, B.; Krager, J.; Putnam, N.; Koraisly, B.; Wood, K.L.; Jensen, D. Design for Transformation: Theory, Method and Application. In Proceedings of the ASME International Design Engineering Technical Conference & Computers and Information in Engineering Conference, Las Vegas, NV, USA, 4–7 September 2007; pp. 447–459.
28. Weaver, J.; Wood, K.L.; Jensen, D. Transformation Facilitators: A Quantitative Analysis of Reconfigurable Products and Their Characteristics. In Proceedings of the International Design Engineering Technical Conferences, Brooklyn, NY, USA, 3–6 August 2008; pp. 351–366.
29. Singh, V. Design for Transformation: Design Principles and Approach with Concept Generation Tools and Techniques. Master's Thesis, University of Texas at Austin, Austin, TX, USA, 2007.
30. Camburn, B.A.; Guillemette, J.; Crawford, R.H.; Jensen, D.J.; Wood, J.J.; Wood, K.L. When to transform? Development of indicators for design context evaluation. In Proceedings of the International Design Engineering Technical Conferences and Computers and Information in Engineering Conference (ASME 2010), Montreal, QC, Canada, 15–18 August 2010; pp. 249–266.
31. Camburn, B.; Wood, K.; Crawford, R.; Robbins, J.; Jensen, D.; Patel, A. Advances in transformational design: Correlating context evaluation to quality feasibility and novelty. In Proceedings of the International Design Engineering Technical Conferences and Computers and Information in Engineering Conference (ASME 2012), Chicago, IL, USA, 12–15 August 2012; pp. 447–460.
32. Son, J. Mechanical Transformation to Support Design for Environmentally Significant Behaviour. Master's Thesis, University of Toronto, Toronto, ON, Canada, 2012.

33. Son, J.; Shu, L.H. The mechanical transformation and environmentally conscious behavior. *AI EDAM* **2014**, *28*, 193–203. [[CrossRef](#)]
34. HaeBin, L. Difficulties in Transformable Design and Its Implications. Master's Thesis, UNIST, Ulsan, Korea, 2016.
35. Li, S. *Introduction to Interaction Design*, 1st ed.; Tsinghua University Press: Beijing, China, 2009.
36. Liu, X. Research on Key Technologies of Situated Design for Innovative Process. Ph.D. Thesis, Hebei University of Technology, Tianjin, China, 2007.
37. DeSouza, K.C.; Awazu, Y.; Jha, S.; Dombrowski, C.; Papagari, S.; Baloh, P.; Kim, J.Y. Customer-driven innovation. *Res. Technol. Manag.* **2008**, *51*, 35–44. [[CrossRef](#)]
38. Hazelrigg, G.A. A framework for decision-based engineering design. *J. Mech. Des.* **1998**, *120*, 653–658. [[CrossRef](#)]
39. Song, W. Requirement management for product-service systems: Status review and future trends. *Comput. Ind.* **2017**, *85*, 11–22. [[CrossRef](#)]
40. Wang, Z.; Chen, C.-H.; Zheng, P.; Li, X.; Khoo, L.P. A novel data-driven graph-based requirement elicitation framework in the smart product-service system context. *Adv. Eng. Inform.* **2019**, *42*, 100983. [[CrossRef](#)]
41. Lowdermilk, T.; Rich, J. *The Customer-Driven Playbook: Converting Customer Feedback into Successful Products*; O'Reilly Media, Inc.: Sebastopol, CA, USA, 2017.
42. Cai, W. Extension theory and its application. *Chin. Sci. Bull.* **1999**, *44*, 1538–1548. [[CrossRef](#)]
43. Terninko, J. *Step-by-Step QFD: Customer-Driven Product Design*, 2nd ed.; Routledge: New York, NY, USA, 1997.
44. Zheng, P.; Xun, X.; Sheng, Q.X. A weighted interval rough number based method to determine relative importance ratings of customer requirements in QFD product planning. *J. Intell. Manuf.* **2019**, *30*, 3–16.
45. Suh, N.P. *Axiomatic Design, Advances and Applications*; Oxford university press: Oxford, UK, 2001.
46. Tang, D.; Zhang, G.; Dai, S. Design as integration of axiomatic design and design structure matrix. *Robot. Comput. Integr. Manuf.* **2009**, *25*, 610–619. [[CrossRef](#)]
47. Farid, A.M.; Luis, R. An axiomatic design of a multiagent reconfigurable mechatronic system architecture. *IEEE Trans. Ind. Inform.* **2015**, *11*, 1142–1155. [[CrossRef](#)]
48. Eilouti, B. Scenario-based design: New applications in metamorphic architecture. *Front. Archit. Res.* **2018**, *7*, 530–543. [[CrossRef](#)]
49. Carroll, J.M. Five reasons for scenario-based design. In Proceedings of the 32nd Hawaii International Conference on System Sciences, Maui, HI, USA, 5–8 January 1999.
50. Suh, N.P. *The Principles of Design*; Oxford University Press on Demand: Oxford, UK, 1990.
51. Otto, K.; Wood, K. *Product Design: Techniques in Reverse Engineering and New Product Development*; Prentice-Hall: Upper Saddle River, NJ, USA, 2001.
52. Pahl, G.; Beitz, W. *Engineering Design: A Systematic Approach*, 2nd ed.; Springer: Berlin, Germany, 2007.
53. Stone, R.B.; Wood, K.L.; Crawford, R.H. A heuristic method for identifying modules for product architectures. *Des. Stud.* **2000**, *21*, 5–31. [[CrossRef](#)]
54. Ulrich, K.T. *Product Design and Development*, 2nd ed.; Tata McGraw-Hill Education: Boston, MA, USA, 2004.
55. Liu, W.; Tan, R.; Cao, G.; Zhang, Z.; Huang, S.; Liu, L. A proposed radicality evaluation method for design ideas at conceptual design stage. *Comput. Ind. Eng.* **2019**, *132*, 141–152. [[CrossRef](#)]
56. Bollobás, B. *Modern Graph Theory*; Springer: New York, NY, USA, 1998.
57. Gao, X.; Xiao, B.; Tao, D.; Li, X. A survey of graph edit distance. *Pattern Anal. Appl.* **2010**, *13*, 113–129. [[CrossRef](#)]
58. Neuhaus, M.; Horst, B. Automatic learning of cost functions for graph edit distance. *Inf. Sci.* **2007**, *177*, 239–247. [[CrossRef](#)]
59. Čech, P. Matching UML class models using graph edit distance. *Expert Syst. Appl.* **2019**, *130*, 206–224. [[CrossRef](#)]
60. Cortés, X.; Donatello, C.; Hubert, C. Learning edit cost estimation models for graph edit distance. *Pattern Recognit. Lett.* **2019**, *125*, 256–263. [[CrossRef](#)]
61. Montani, S.; Leonardi, G.; Quaglini, S.; Cavallini, A.; Micieli, G. A knowledge-intensive approach to process similarity calculation. *Expert Syst. Appl.* **2015**, *42*, 4207–4215. [[CrossRef](#)]
62. Zhang, S.; Zheng, X.; Hu, C. A survey of semantic similarity and its application to social network analysis. In Proceedings of the 2015 IEEE International Conference on Big Data, Santa Clara, CA, USA, 29 October–1 November 2015.

63. Tian, J.; Zhao, W. Words Similarity Algorithm Based on Word Tree in Semantic Web Adaptive Learning System. *J. Univ. Inf. Sci. Ed.* **2010**, *28*, 602–608.
64. Semantic Similarity Computing and Programming Based on Synonym Forest. Available online: <http://download.csdn.net/download/huandaohack/4557989> (accessed on 5 December 2019).
65. Zhang, H. *Innovative Design-Systematic Innovation Based on TRIZ*; Machinery Industry Press: Beijing, China, 2017.
66. Wheelchair. Available online: <http://wikipedia.moesalih.com/Wheelchair> (accessed on 5 December 2019).



© 2020 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 (<http://creativecommons.org/licenses/by/4.0/>).