applied sciences

MDPI

Article

Extension Design Model of Rapid Configuration Design for Complex Mechanical Products Scheme Design

Tichun Wang *, Hao Li and Xianwei Wang

College of Mechanical and Electrical Engineering, Nanjing University of Aeronautics and Astronautics, Nanjing 211106, China
* Correspondence: wangtichun2010@nuaa.edu.cn

Abstract: This study explores the extension configuration methods of complex product conceptual design, seeking to improve the product design efficiency and design quality. The paper firstly reviews the literature on element representation models of multi-type design knowledge, followed by a review on extension design models for the rapid configuration of complex product conceptual design. The extension transformation method for the rapid configuration design of complex product conceptual design is also reviewed. With the analysis of the extension reasoning model for the rapid configuration design of complex product conceptual design, the research proposes a new model of extension reasoning for the rapid configuration design of complex product conceptual design. This model of extension design would enhance the rapid configuration design and conceptual design of large and complex products. Detailed steps of the algorithm implementation are also presented. This study also tests the validity and operability of the model and the algorithm with the design case of a large hydro-turbine product design.

Keywords: intelligent design; knowledge engineering; extension theory; knowledge reasoning; knowledge reuse

check for updates

Citation: Wang, T.; Li, H.; Wang, X. Extension Design Model of Rapid Configuration Design for Complex Mechanical Products Scheme Design. *Appl. Sci.* **2022**, *12*, 7921. https://doi.org/10.3390/app12157921

Academic Editors: Božidar Šarler, Haiping Liu, Jian Zhang and Jiaqi Li

Received: 27 June 2022
Accepted: 25 July 2022
Published: 7 August 2022

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

Copyright: © 2022 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (https://creativecommons.org/licenses/by/4.0/).

1. Introduction

Generally, the process of complex product scheme design is a complex, multi-level, multi-attribute and creative design configuration process, in which there are many types of design knowledge, design constraints and design conflicts. Therefore, it is of great engineering application value to summarize the various types of design knowledge in complex product scheme design and reuse the previous design knowledge in new scheme designs after a series of knowledge acquisition, expression, storage and reuse [1–3]. For this reason, many scholars have studied the intelligent configuration design of large and complex products from different perspectives [4–9]. However, relevant studies pay more attention to the processing of surface knowledge and the analysis of configuration design results, ignoring the processing of the configuration process and in-depth knowledge, which plays an important role in improving the configuration design ability of complex product design schemes.

Extenics is a new discipline that uses formal models to study the possibility of expanding things and the laws and methods of development and innovation. It uses formal tools to study the laws and methods of contradictory problems from the qualitative and quantitative perspectives. It takes primitive theory and extension mathematics as the pillars and formalizes the process of problem solving, so as to establish corresponding mathematical models and develop new calculation methods and technologies. It can more intelligently and formally solve the storage, representation and processing of deep knowledge in the knowledge base, and promote the knowledge in knowledge engineering in a more formal, deeper and more essential direction [10–12]. Therefore, studying the application of Extenics in the intelligent configuration design of complex products is of great significance to further

solve the above problems in the intelligent configuration design of complex products and improve the research methods regarding the efficiency of the intelligent configuration design of complex products [13–15].

At present, Extenics has some successful applications in the field of product configuration design [16–18], but the systematic application of Extenics in the configuration design of complex product design schemes is rare, and there are still some problems that need to be solved urgently, such as how to further promote the research on the results of configuration design to the research on the process of configuration design, how to solve the expression, storage and application problems of various types of deep knowledge in the process of scheme design, how to solve the extension reasoning and extension reuse problems in the process of solution design, etc.

Therefore, based on the existing research results [19–23], this paper studies the extension design pattern of the rapid configuration of complex product scheme design, and puts forward the knowledge model, extension transformation model, extension reasoning model and extension reuse model for the extension configuration of complex product scheme design.

2. Extension Modeling in Conceptual Design Extension Design Mode

Complex product conceptual design is a multi-level, multi-attribute and creative product structure configuration process, in which there are many types of domain design knowledge, including empirical knowledge, design rules and criteria, principal knowledge, structural knowledge, process knowledge, etc., and it has various forms of expression, such as formulas, diagrams, tables and examples. Therefore, to deal with the various complex design reasoning problems in product conceptual design, it is necessary to solve the storage, representation and processing of deep knowledge in the process of conceptual design reasoning. To this end, the primitive theory in Extenics is introduced into the product conceptual design. It uses primitives as the logical cells of extension design, and unifies the design features that characterize the quality and quantity, actions and relationships of the design objects in a design object Γ, design feature c and the feature value v of design object Γ regarding design feature c in the ordered triple $\boldsymbol{J} = (\Gamma, c, v)$. From a qualitative and quantitative perspective, the information, behavior and relationships in the design process are described in a formalized and modeled manner, which provides a new methodological system for individuals to understand the real world and resolve contradictions and reasoning in the real world.

This paper divides the design information in conceptual design into static design information, behavioral design information and relational design information.

2.1. Matter-Element Model

When modeling static design information, the matter-element model $\boldsymbol{J}(\boldsymbol{R})$ in the primitive theory is used for description. If the described design object has design features, the matter-element model $\boldsymbol{J}(\boldsymbol{R})$ is expressed as follows:

$$\boldsymbol{J}(\boldsymbol{R}) = \begin{bmatrix} \Gamma(N) & C(N)_1 & [V(C)_1, W(C)_1] \\ & C(N)_2 & [V(C)_2, W(C)_2] \\ & \vdots & \vdots \\ & C(N)_n & [V(C)_n, W(C)_n] \end{bmatrix} \tag{1}$$

Among them, $\Gamma(N)$ is the name of the description object; $V(C)$ is the magnitude of the design feature; $W(C)$ is the weight of the design feature, and $V(C)$ and $W(C)$ can be precise point values, interval magnitudes with fuzzy information, membership functions and qualitative semantic descriptions and other forms. Therefore, to express more generality,

assuming $V = [v^L, v^R]$, $W = [w^L, w^R]$, both are forms of interval information with fuzzy characteristics. Thus, Formula (1) can be expressed as follows:

$$\boldsymbol{J}(\boldsymbol{R}) = \begin{bmatrix} \Gamma(N) & C(N)_1 & \left([v(C)_1^L, v(C)_1^R], [w(C)_1^L, w(C)_1^R]\right) \\ & C(N)_2 & \left([v(C)_2^L, v(C)_2^R], [w(C)_2^L, w(C)_2^R]\right) \\ & \vdots & \vdots \\ & C(N)_n & \left([v(C)_n^L, v(C)_n^R], [w(C)_n^L, w(C)_n^R]\right) \end{bmatrix} \tag{2}$$

2.2. Event-Element Model

When modeling behavior-based design information, the event-element model $\boldsymbol{J}(\boldsymbol{I})$ in the primitive theory is used to describe it. If the described design behavior has design features, the event-element model $\boldsymbol{J}(\boldsymbol{I})$ is expressed as follows:

$$\boldsymbol{J}(\boldsymbol{I}) = \begin{bmatrix} \Gamma(D) & B(D)_1 & \left(U(B)_1, [w(B)_1^L, w(B)_1^R]\right) \\ & B(D)_2 & \left(U(B)_2, [w(B)_2^L, w(B)_2^R]\right) \\ & \vdots & \vdots \\ & B(D)_m & \left(U(B)_m, [w(B)_m^L, w(B)_m^R]\right) \end{bmatrix} \tag{3}$$

Among them, $\Gamma(D)$ is the term describing the design behavior; $B(D)$ is the operation feature of the design behavior, and $W(B)$ is the decision weight when the operation feature is executed.

2.3. Relational Meta-Model

When modeling relational design information, the relational meta-model $\boldsymbol{J}(\boldsymbol{Q})$ in the primitive theory is used to describe the configuration relation, logical relation, implication relation, comparison relation, assembly relation, etc. If the described design constraint relationship has design features, the relational meta-model $\boldsymbol{J}(\boldsymbol{Q})$ is expressed as follows:

$$\boldsymbol{J}(\boldsymbol{Q}) = \begin{bmatrix} \Gamma(S) & A(S)_1 & \left(G(A)_1, [w(A)_1^L, w(A)_1^R]\right) \\ & A(S)_2 & \left(G(A)_2, [w(A)_2^L, w(A)_2^R]\right) \\ & \vdots & \vdots \\ & A(S)_k & \left(G(A)_k, [w(A)_k^L, w(A)_k^R]\right) \end{bmatrix} \tag{4}$$

Among them, $\Gamma(S)$ is the term describing the design constraint relationship; $A(S)$ is the associated feature of the design constraint relationship, and $W(A)$ is the associated degree of the associated feature.

In the conceptual design process of complex products, design knowledge often has the quality of mixed characteristics—that is, the combination of static design information, design behavior and design constraint relationships. To this end, the composite meta-model $\boldsymbol{J}(\boldsymbol{F})$ in the primitive theory is used to model it, and the function of the connective Θ is used to represent the design information with multi-layered semantics and more abundant design information. The commonly used connective has the "$\wedge$" or the conjunction "$\vee$", etc., to generate the corresponding compound element or the compound element, etc., and then form the overall design information of the scheme design. The composite meta-model $\boldsymbol{J}(\boldsymbol{F})$ can be represented as follows:

$$\boldsymbol{J}(\boldsymbol{F}) = \begin{bmatrix} \Gamma(F) & (\Theta)\Gamma(\boldsymbol{J}(\boldsymbol{R}_i)) & (V(\boldsymbol{J}(\boldsymbol{R}_i)), W(\boldsymbol{J}(\boldsymbol{R}_i))) \\ & (\Theta)\Gamma(\boldsymbol{J}(\boldsymbol{I}_j)) & (V(\boldsymbol{J}(\boldsymbol{I}_j)), W(\boldsymbol{J}(\boldsymbol{I}_j))) \\ & (\Theta)\Gamma(\boldsymbol{J}(\boldsymbol{Q}_s)) & (V(\boldsymbol{J}(\boldsymbol{Q}_s)), W(\boldsymbol{J}(\boldsymbol{Q}_s))) \end{bmatrix} \tag{5}$$

Among them, i, j, s represent the number of matter elements, event elements and relation elements contained in the compound element, respectively.

It should be noted here that if the above-mentioned matter-element model $\boldsymbol{J}(\boldsymbol{R})$, event-element model $\boldsymbol{J}(\boldsymbol{I})$, relational meta-model $\boldsymbol{J}(\boldsymbol{Q})$ and composite meta-model $\boldsymbol{J}(\boldsymbol{F})$ are used

to model design knowledge, when it only expresses the state of existence of the design attribute, but does not reflect the importance of the design attribute, the weight value in the above primitive model does not need to be included. The extensibility of design objects provides effective support for the extension and reasoning of design reuse in the conceptual design process.

3. Extension Design Pattern for Conceptual Design and Rapid Configuration

Product configuration design is one of the most important design methods to achieve mass customization, which is driven by customer needs and runs through the entire product design process. Therefore, to quickly configure and design complex products, it is necessary to effectively plan and quickly combine subsequent product design parameters based on the analysis and mapping of product conceptual design requirements. At present, there has been much research on rapid product configuration design at home and abroad, and corresponding research results have been obtained. However, the existing product configuration design methods have the following main problems that limit their development in knowledge reasoning, knowledge reuse and knowledge mining in complex product design. (1) The current product configuration design method lacks a consistent description of configuration design problems that combines quantitative and qualitative features. If the configuration design features provided are few, it will be difficult to generate products through local configuration design features, which will affect the subsequent rapid product design work. (2) In the process of product configuration design, a large number of configuration variables are often generated. Most of the existing knowledge models and knowledge reasoning methods are based on the reasoning of superficial knowledge to analyze the convergence of configuration variables and solve the problem of configuration contradictions. Insufficient extended reasoning support affects the effect of deep knowledge processing, thus limiting the scope of design extension reuse.

The research method of this paper is the extension intelligent design method. The implementation of this method includes four key links: knowledge modeling, knowledge extension transformation, knowledge extension reasoning and knowledge extension reuse. The knowledge modeling link deals with the formalization and modeling of various types of knowledge. Based on the existing knowledge model, the knowledge extension transformation link can obtain rich design knowledge through expansion and transformation, and then support the ability of knowledge extension reasoning. The knowledge extension reasoning link mainly solves the reasoning problem in the extension configuration process of the design scheme, and obtains the object that can match the design goal. Knowledge extension reuse mainly solves the problem of the extension reuse of existing design objects, and realizes the reuse of the optimal objects in the extension configuration design of products. The four extension design links are integrated into an organic whole, so as to realize the extension configuration design of complex product design schemes.

Therefore, this paper will first present the extension transformation method in rapid configuration design to effectively plan and expand the configuration design, and then introduce the extension distance and construct the extension correlation function, and propose a product rapid configuration design method integrating design process reasoning and design case reasoning. Based on the effective division of product design parameters, an extension reuse method for product rapid configuration design is proposed. Finally, the implementation framework of the extension design pattern for the rapid configuration design of complex products is given.

3.1. Extension Transformation Model for Rapid Configuration Design

The rapid configuration design based on primitive theory adopts a consistent configuration design problem description combining quantitative aspects, expands the design information in the field based on the primitive model of the description object and obtains more design knowledge to support the rapid configuration of products. Aiming at the structural characteristics of the knowledge primitive model of complex product conceptual

design, based on the extension theory, the design knowledge expansion is carried out by means of extensible transformation—that is, the three aspects of the design universe, design characteristics and design value of the primitive model can be expanded. The specific implementation method is described as follows.

(1) Transform the feature value of the design object primitive in the process of product rapid configuration design, and realize the extension transformation of the scalable design information, i.e.,

$$\boldsymbol{J}_{T(V)} = \left\{ (\boldsymbol{J}^*, K_V, K_V^*) | \boldsymbol{J}^* \in (\boldsymbol{J} \to \boldsymbol{J}_i)_{T(V)}, K_V = K(v(\boldsymbol{J})), K_V^* = K^* \left(v(\boldsymbol{J} \to \boldsymbol{J}_i)_{T(V)} \right) \right\} \quad (6)$$

Among them, $T(V)$ is the extension transformation of the feature value of the primitive $\boldsymbol{J}$ of the process innovation scalable information object of manufacturing enterprises; K_V, K_V^* represent, respectively, the feature association of the feature value transformation of the design object primitive $\boldsymbol{J}$ in the process of product rapid configuration design function and extension function; K_V is a measure of whether the feature value of the design object primitive J can be extended, and K_V^* is a measure of the extension degree of the feature value of the design object primitive $\boldsymbol{J}$.

(2) Transform the primitive features of the design object primitive in the process of product rapid configuration design to realize the feature extension transformation of scalable design information, i.e.,

$$\boldsymbol{J}_{T(C)} = \left\{ (\boldsymbol{J}^*, K_C, K_C^*) | \boldsymbol{J}^* \in (\boldsymbol{J} \to \boldsymbol{J}_i)_{T(C)}, K_C = K(c(\boldsymbol{J})), K_C^* = K^* \left(c(\boldsymbol{J} \to \boldsymbol{J}_i)_{T(C)} \right) \right\} \quad (7)$$

Among them, $T(C)$ is the extension transformation of the primitive features of the primitive $\boldsymbol{J}$ of the process innovation extensible information object of manufacturing enterprises; K_C, K_C^* represent, respectively, the feature association of the primitive feature transformation of the design object primitive $\boldsymbol{J}$ in the process of product rapid configuration design function and extension function; K_C is a measure of whether the design feature of the design object primitive $\boldsymbol{J}$ can be extended, and K_C^* is a measure of the extension degree of the design feature of the design object primitive $\boldsymbol{J}$.

(3) Transform the design domain of the design object primitive in the process of product rapid configuration design, and realize the extension transformation of different domain constraints of scalable design information, i.e.,

$$\boldsymbol{J}_{T(\Omega)} = \left\{ (\boldsymbol{J}^*, K_\Omega, K_\Omega^*) | \boldsymbol{J}^* \in (\boldsymbol{J} \to \boldsymbol{J}_i)_{T(\Omega)}, K_\Omega = K(\Omega(\boldsymbol{J})), K_\Omega^* = K^* \left(\Omega(\boldsymbol{J} \to \boldsymbol{J}_i)_{T(\Omega)} \right) \right\} \quad (8)$$

Among them, $T(\Omega)$ is the extension transformation of the design universe of the process innovation extensible information object primitive $\boldsymbol{J}$ of manufacturing enterprises; K_Ω, K_Ω^* represent, respectively, the feature association of the design universe transformation of the design object primitive $\boldsymbol{J}$ in the process of product rapid configuration design function and extension function; K_Ω is a measure of whether the design universe of the design object primitive $\boldsymbol{J}$ can be extended, and K_Ω^* is a measure of the extension degree of the design universe of the design object primitive $\boldsymbol{J}$.

Whether the extension transformation of rapid configuration design can be effectively applied to product design needs to follow the rules of extension configuration—that is, it needs to meet the requirements of the association degree and extension degree before it can be used as a configuration design reuse object and applied to subsequent designs.

The rules for the configuration of the magnitude extension of the design primitive $\boldsymbol{J}$:

$$if\ (K(v(\boldsymbol{J})) \geq 0) \wedge \left(K^* \left(v(\boldsymbol{J} \to \boldsymbol{J}_i)_{T(V)} \right) \geq K_0(V) \right)\ then\ (\boldsymbol{J}^* = T(V)\boldsymbol{J}) \in \widetilde{\boldsymbol{S}}_{J(T)} \quad (9)$$

The rules for feature extension configuration of design primitive $\boldsymbol{J}$:

$$if\ (K(c(\boldsymbol{J})) \geq 0) \wedge \left(K^* \left(c(\boldsymbol{J} \to \boldsymbol{J}_i)_{T(C)} \right) \geq K_0(C) \right)\ then\ (\boldsymbol{J}^* = T(C)\boldsymbol{J}) \in \widetilde{\boldsymbol{S}}_{J(T)} \quad (10)$$

The rules for the extension configuration of the design universe of the design primitive $\boldsymbol{J}$:

$$if\ (K(\Omega(\boldsymbol{J})) \geq 0) \wedge \left(K^*\left(\Omega(\boldsymbol{J} \to \boldsymbol{J}_i)_{T(\Omega)}\right) \geq K_0(\Omega)\right)\ then\ (\boldsymbol{J}^* = T(\Omega)\boldsymbol{J}) \in \widetilde{\boldsymbol{S}}_{J(T)} \tag{11}$$

Among them, $K_0(V)$, $K_0(C)$, $K_0(\Omega)$ represent the threshold of the extension degree, respectively, and $\widetilde{\boldsymbol{S}}_{J(T)}$ represents the extension stability domain of the primitive obtained by the extension transformation—that is, the extension reuse set of the design object.

In particular, if the feature of the design object primitive, the feature value and the correlation degree of the design universe are all greater than zero, and there is

$$\left(0 \leq K^*\left(v(\boldsymbol{J} \to \boldsymbol{J}_i)_{T(V)}\right) \leq K_0(V)\right) \vee \left(0 \leq K^*\left(c(\boldsymbol{J} \to \boldsymbol{J}_i)_{T(C)}\right) \leq K_0(C)\right) \vee \left(0 \leq K^*\left(\Omega(\boldsymbol{J} \to \boldsymbol{J}_i)_{T(\Omega)}\right) \leq K_0(\Omega)\right)$$

then $(\boldsymbol{J}^* = T(V)\boldsymbol{J}) \vee (\boldsymbol{J}^* = T(C)\boldsymbol{J}) \vee (\boldsymbol{J}^* = T(\Omega)\boldsymbol{J}) \in \widetilde{\boldsymbol{S}}^*_{J(T)}$, $\widetilde{\boldsymbol{S}}^*_{J(T)}$ is the extension domain of the primitives obtained by extension transformation, which means that although the primitives in $\widetilde{\boldsymbol{S}}^*_{J(T)}$ cannot be used for design reuse, they can be used as the extension objects of the subsequent design of the product or in the design of new products in the field. The number of expandable objects will increase, so that richer knowledge of the rapid configuration design can be generated, and the reasoning ability and knowledge reuse of the product rapid configuration design can be improved.

3.2. Extension Reasoning Model for Rapid Configuration Design

The essence of the rapid configuration design process of complex products based on the primitive model is to realize the effective matching and retrieval of primitives in the configuration design process. In the process of primitive modeling, a knowledge hierarchy is used to design a knowledge association, so the hierarchical retrieval strategy of classification storage and classification retrieval is used to match and retrieve primitives with modeled and formalized features. For a given target primitive, it is necessary to perform primitive matching in the knowledge base based on the matching features. The target primitive can be a requirement primitive or a design object primitive transformed according to the design's technical requirements or obtained through extension transformation. For the generality of the expression, this article will collectively refer to it as the target primitive $\boldsymbol{J}_0$, and the target primitive corresponds to the existing design object primitives in the knowledge base, the instance base and the rule base, which are referred to as design primitives $\boldsymbol{J}_d$. The extension matching between the target primitive $\boldsymbol{J}_0$ and the design primitive $\boldsymbol{J}_d$ has two meanings: one is to match the primitive features of the target primitive $\boldsymbol{J}_0$ and the design primitive $\boldsymbol{J}_d$ and the other is to analyze the extension association of the primitive features.

Assuming that the number of primitive features of the target primitive $\boldsymbol{J}_0$ is N_o, the number of primitive features of the design primitive $\boldsymbol{J}_d$ is N_d and the number of primitive features of the two common primitives is N_k, then the primitive feature difference Δ_{J0} of the target primitive $\boldsymbol{J}_0$ is:

$$\Delta_{\boldsymbol{J}_o} = N_o - N_k \tag{12}$$

The primitive feature difference Δ_{Jd} of the design primitive $\boldsymbol{J}_d$ is:

$$\Delta_{\boldsymbol{J}_d} = N_d - N_k \tag{13}$$

The common primitive feature difference Δ_J of target primitive Δ_{J0} and design primitive Δ_{Jd} is:

$$\Delta_{\boldsymbol{J}} = \Delta_{\boldsymbol{J}_o} + \Delta_{\boldsymbol{J}_d} = N_o + N_d - 2N_k \tag{14}$$

Then, the basic feature matching coefficient of target primitive $\boldsymbol{J}_0$ and design primitive $\boldsymbol{J}_d$ is φ_J:

$$\varphi_{\boldsymbol{J}} = 1 - (N_o + N_d - 2N_k)/(N_o + N_d - N_k) \tag{15}$$

It can be seen that $0 \le \varphi_J \le 1$, when $\varphi_J = 1$, it means that the features of the target primitive J_0 and the design primitive J_d are completely matched, and when $\varphi_J = 0$, it means that the target primitive J_0 and the design primitive J_d features are completely mismatched.

Only the design primitive J_d that matches the primitive feature can be used in the subsequent design. Therefore, it is necessary to perform an association analysis on the matched primitive features. Due to the use of primitive modeling, the eigenvalues of the target primitive J_0 are described by a combination of qualitative and quantitative descriptions, and the qualitatively described eigenvalues can be converted into the expression form of fuzzy numbers. Therefore, for convenience of expression, the eigenvalues of the target primitive J_0 and the design primitive J_d are both expressed in the form of fuzzy intervals—that is, $v(J_o) = \left[v^l_{(J_o)}, v^r_{(J_o)}\right]$, $v(J_d) = \left[v^l_{(J_d)}, v^r_{(J_d)}\right]$.

If the common primitive feature of target primitive J_0 and design primitive J_d is an intermediate feature—that is, the optimal value of the feature value is the midpoint of the interval—then the extension distance of the common primitive feature of target primitive J_0 and design primitive J_d for $\rho(v(J_d), v(J_o))$ is:

$$\begin{aligned}\rho(v(J_d), v(J_o)) &= \rho\left(\left[v^l_{(J_o)}, v^r_{(J_o)}\right], \left[v^l_{(J_d)}, v^r_{(J_d)}\right]\right) \\ &= \left(\left(\left|v^l_{(J_d)} - \frac{v^r_{(J_o)}+v^l_{(J_o)}}{2}\right| - \frac{1}{2}\left(v^r_{(J_o)} - v^l_{(J_o)}\right)\right) + \left(\left|v^r_{(J_d)} - \frac{v^r_{(J_o)}+v^l_{(J_o)}}{2}\right| - \frac{1}{2}\left(v^r_{(J_o)} - v^l_{(J_o)}\right)\right)\right)/2\end{aligned} \tag{16}$$

If the common primitive feature of target primitive J_0 and design primitive J_d is a positive feature—that is, the optimal value $v^*(J_0)$ of the feature value is located on the right side of the interval—then the common primitive feature of target primitive J_0 and design primitive J_d can be the right side distance is $\rho^R(v(J_d), v(J_o))$:

$$\rho^R\left(v^l_{(J_d)}, v(J_o)\right) = \begin{cases} v^l_{(J_o)} - v^l_{(J_d)}, & v^l_{(J_d)} \le v^*(J_o) \\ \frac{v^l_{(J_o)} - v^*(J_o)}{v^r_{(J_o)} - v^*(J_o)} * \left(v^r_{(J_o)} - v^l_{(J_d)}\right), & v^*(J_o) \le v^l_{(J_d)} \le v^r_{(J_o)} \\ v^l_{(J_d)} - v^r_{(J_o)}, & v^l_{(J_d)} \ge v^r_{(J_o)} \end{cases} \tag{17}$$

$$\rho^R\left(v^r_{(J_d)}, v(J_o)\right) = \begin{cases} v^l_{(J_o)} - v^r_{(J_d)}, & v^r_{(J_d)} \le v^*(J_o) \\ \frac{v^l_{(J_o)} - v^*(J_o)}{v^r_{(J_o)} - v^*(J_o)} * \left(v^r_{(J_o)} - v^r_{(J_d)}\right), & v^*(J_o) \le v^r_{(J_d)} \le v^r_{(J_o)} \\ v^r_{(J_d)} - v^r_{(J_o)}, & v^r_{(J_d)} \ge v^r_{(J_o)} \end{cases} \tag{18}$$

$$\rho^R(v(J_d), v(J_o)) = \left(\rho^R\left(v^l_{(J_d)}, v(J_o)\right) + \rho^R\left(v^r_{(J_d)}, v(J_o)\right)\right)/2 \tag{19}$$

If the common primitive feature of target primitive J_0 and design primitive J_d is a reverse-type feature—that is, the optimal value $v^*(J_0)$ of the feature value is located on the left side of the interval—then the extension of the common primitive feature of target primitive J_0 and design primitive J_d can be the left distance if $\rho^L(v(J_d), v(J_o))$:

$$\rho^L\left(v^l_{(J_d)}, v(J_o)\right) = \begin{cases} v^l_{(J_o)} - v^l_{(J_d)}, & v^l_{(J_d)} \le v^l_{(J_o)} \\ \frac{v^r_{(J_o)} - v^*(J_o)}{v^l_{(J_o)} - v^*(J_o)} * \left(v^l_{(J_d)} - v^l_{(J_o)}\right), & v^l_{(J_o)} \le v^l_{(J_d)} \le v^*(J_o) \\ v^l_{(J_d)} - v^r_{(J_o)}, & v^l_{(J_d)} \ge v^*(J_o) \end{cases} \tag{20}$$

$$\rho^L\left(v^r_{(J_d)}, v(J_o)\right) = \begin{cases} v^l_{(J_o)} - v^r_{(J_d)}, & v^r_{(J_d)} \le v^l_{(J_o)} \\ \frac{v^r_{(J_o)} - v^*(J_o)}{v^l_{(J_o)} - v^*(J_o)} * \left(v^r_{(J_d)} - v^l_{(J_o)}\right), & v^l_{(J_o)} \le v^r_{(J_d)} \le v^*(J_o) \\ v^r_{(J_d)} - v^r_{(J_o)}, & v^r_{(J_d)} \ge v^*(J_o) \end{cases} \tag{21}$$

$$\rho^L(v(J_d), v(J_o)) = \left(\rho^L\left(v^l_{(J_d)}, v(J_o)\right) + \rho^L\left(v^r_{(J_d)}, v(J_o)\right)\right)/2 \tag{22}$$

Since different primitive feature values often have different dimensions, in order to correctly perform design extension inference, it is necessary to normalize the extension distance. It is assumed that the number of design primitives corresponding to the common primitive feature i is m; then, the extension distance $\overline{\rho_{ij}}(v(\boldsymbol{J}_d), v(\boldsymbol{J}_o))$ after normalization is:

$$\overline{\rho_{ij}}(v(\boldsymbol{J}_d), v(\boldsymbol{J}_o)) = \begin{cases} -\frac{\rho_{ij}(v(\boldsymbol{J}_d), v(\boldsymbol{J}_o))}{\max\limits_{1\le i\le m} \rho_{ij}(v(\boldsymbol{J}_d), v(\boldsymbol{J}_o))}, & \rho_{ij}(v(\boldsymbol{J}_d), v(\boldsymbol{J}_o)) > 0 \\ 0, & \rho_{ij}(v(\boldsymbol{J}_d), v(\boldsymbol{J}_o)) = 0 \\ -\frac{\rho_{ij}(v(\boldsymbol{J}_d), v(\boldsymbol{J}_o))}{\max\limits_{1\le i\le m} |\rho_{ij}(v(\boldsymbol{J}_d), v(\boldsymbol{J}_o))|}, & \rho_{ij}(v(\boldsymbol{J}_d), v(\boldsymbol{J}_o)) < 0 \end{cases} \tag{23}$$

3.3. Extension Reuse Model for Rapid Configuration Design

The core of rapid configuration design extension reuse is to reuse the primitive extension set or the extension reuse set of design object primitives obtained based on extension transformation in the design of new solutions through extension reasoning and analysis. In the process of designing object extension and reuse, it is necessary to maintain the extension correlation of primitive features—that is, the matching, correlation and similarity between primitive features need to be considered. The mathematical model of the extension reuse process for rapid product configuration design can be expressed as follows:

$$\boldsymbol{E}_{EKR} = \left[\widetilde{\boldsymbol{J}}_o^{RS}, \widetilde{\boldsymbol{J}}_d^{RS}, \Phi\left(\widetilde{\boldsymbol{J}}_o^{RS} \to \widetilde{\boldsymbol{J}}_d^{RS}\right)\right] \tag{24}$$

Among them, J_0^{RS} represents the extension set of finite target primitives $\boldsymbol{J}_0$, namely demand primitives; J_d^{RS} represents the extension set of finite design primitives $\boldsymbol{J}_d$, and $\Phi\left(\widetilde{\boldsymbol{J}}_o^{RS} \to \widetilde{\boldsymbol{J}}_d^{RS}\right)$ represents the extension correlation function between target primitive $\boldsymbol{J}_0$ and design primitive $\boldsymbol{J}_d$.

The extension correlation function should be able to represent the matching, correlation and similarity between primitive features, and it can be expressed as:

$$\Phi\left(\widetilde{\boldsymbol{J}}_o^{RS} \to \widetilde{\boldsymbol{J}}_d^{RS}\right) = \varphi\left(\widetilde{\boldsymbol{J}}_o^{RS} \to \widetilde{\boldsymbol{J}}_d^{RS}\right) \otimes \lambda\left(\widetilde{\boldsymbol{J}}_o^{RS} \to \widetilde{\boldsymbol{J}}_d^{RS}\right) \otimes \sigma\left(\widetilde{\boldsymbol{J}}_o^{RS} \to \widetilde{\boldsymbol{J}}_d^{RS}\right) \tag{25}$$

$\varphi\left(\widetilde{\boldsymbol{J}}_o^{RS} \to \widetilde{\boldsymbol{J}}_d^{RS}\right)$ is the primitive feature matching function between the target primitive $\boldsymbol{J}_0$ and the design primitive $\boldsymbol{J}_d$, which is used to describe the primitive feature matching characteristics, and is defined as:

$$\varphi : \left(\boldsymbol{J}_o \in \widetilde{\boldsymbol{J}}_o^{RS}\right) \times \left(\boldsymbol{J}_d \in \widetilde{\boldsymbol{J}}_d^{RS}\right) \to K_\varphi(\boldsymbol{J}_d) \tag{26}$$

Among them, the meaning of this formula is equivalent to Formula (25)—that is, the matching degree $K_\varphi(\boldsymbol{J}_d) = \varphi_J$ of the primitive features of the design primitive $\boldsymbol{J}_d$ that meets the requirements.

$\lambda\left(\widetilde{\boldsymbol{J}}_o^{RS} \to \widetilde{\boldsymbol{J}}_d^{RS}\right)$ is the primitive feature correlation function between the target primitive $\boldsymbol{J}_0$ and the design primitive $\boldsymbol{J}_d$, which is used to describe the primitive feature correlation between the design primitive $\boldsymbol{J}_d$ and the target primitive $\boldsymbol{J}_0$, and is defined as:

$$\lambda : \left(\boldsymbol{J}_o \in \widetilde{\boldsymbol{J}}_o^{RS}\right) \times \left(\boldsymbol{J}_d \in \widetilde{\boldsymbol{J}}_d^{RS}\right) \to K_\lambda(\boldsymbol{J}_d) \tag{27}$$

Among them, $K_\lambda(\boldsymbol{J}_d)$ represents the correlation degree of the primitive features of the design primitive $\boldsymbol{J}_d$ that meets the requirements. In this paper, the importance of the primitive features of the design primitive $\boldsymbol{J}_d$ relative to the design requirements—that is, the weight—is taken as the correlation value between the design primitive $\boldsymbol{J}_d$ and the target primitive $\boldsymbol{J}_0$.

$\sigma\left(\widetilde{J}_o^{RS} \to \widetilde{J}_d^{RS}\right)$ is the primitive feature similarity function between the target primitive J_0 and the design primitive J_d, which is used to describe the primitive feature similarity between the design primitive J_d and the target primitive J_0, and is defined as:

$$\sigma : \left(J_o \in \widetilde{J}_o^{RS}\right) \times \left(J_d \in \widetilde{J}_d^{RS}\right) \to K_\sigma(J_d) \tag{28}$$

where $K_\sigma(J_d)$ represents the similarity of the features of the design primitive J_d that meets the requirements, which can be obtained based on the extension distance form discussed in Section 3.2.

In summary, the extension correlation function between primitive features is expressed as:

$$\Phi : K_\varphi(J_d) \otimes K_\lambda(J_d) \otimes K_\sigma(J_d) \to K_\Phi(J_d) \tag{29}$$

If the number of common features of the target primitive J_0 and the design primitive J_d is N_k, the extension reutilization $D(J_d)$ of the design primitive J_d is:

$$D(J_d) = D(J_d \to J_o) = \sum_{i=1}^{N_k} \Phi\left(\widetilde{J}_{oi}^{RS} \to \widetilde{J}_{di}^{RS}\right) = \sum_{i=1}^{N_k} K_\Phi(J_{di}) \tag{30}$$

If the primitive model J_d is in the form of a composite element—that is, it has the characteristics of a hierarchical structure—the extension reuse $D_F(J_d)$ of the composite element J_d is:

$$D_F(J_d) = D_F(J_d \to J_o) = \sum_{i=1}^{N_d} \left(w_{J_{di}} * D(J_{di})\right) \tag{31}$$

Among them, J_{di} is the primitive contained in the compound element J_d, which can be any form of matter element, event element and relation element; N_d is the number of primitives contained in the compound element J_d; w_{di} is the corresponding primitive weight, and satisfies $\sum_{i=1}^{N_d} w_{J_{di}} = 1$.

If the obtained extension reutilization $D(J_d)$ of the design element J_d or the extension reutilization $D_F(J_d)$ of the composite element J_d satisfies the requirement of the reutilization threshold, it means that the design element J_d meets the design requirements, and it can be reused for the extension of the product design.

3.4. Implementation of Extension Design Patterns for Rapid Configuration Design

In the process of rapid configuration and design of complex products, the target primitives of product design are obtained according to the design requirements, matched with the design primitives in the primitive knowledge base. The design primitives that meet the matching requirements will be reused in the new product design. Design primitives that meet the design requirements will not be reused, or the extension analysis and transformation of the primitives will be used to make them meet the requirements. The extension design mode of rapid product configuration design can obtain rich design knowledge in the field based on the extension transformation method, realize the matching of design objects through extension inference model and reuse the best-matched design objects as new ones through the extension reuse model. The design of the product, therefore, will be able to provide support for the smooth implementation of the extension design of complex products. Figure 1 shows the framework for implementing the extension design pattern for rapid product configuration design.

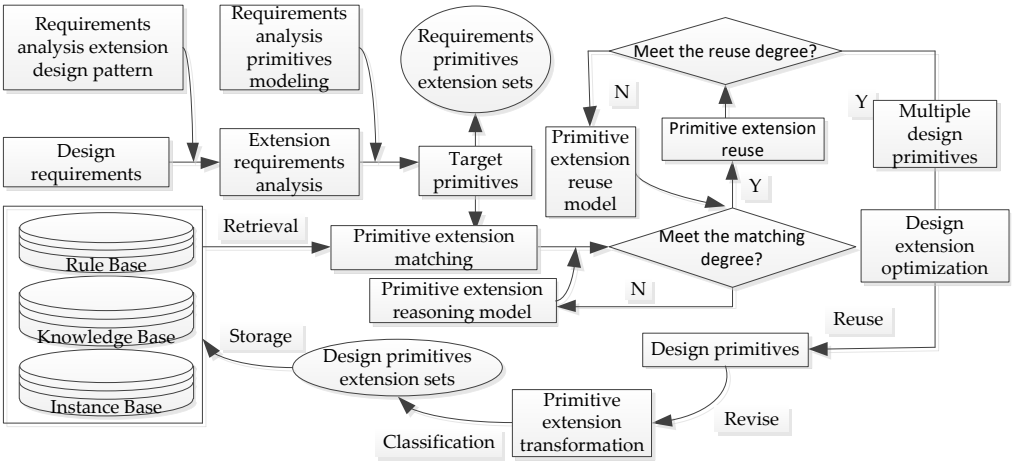


Figure 1. Product conceptual design requirements analysis for extension design patterns.

In summary, the implementation steps of the extension design pattern of rapid product configuration design are described as follows.

Step 1: Based on the extension requirement analysis, the target primitives of product design in the design field are obtained, and the corresponding primitive extension sets are established.

Step 2: Build the extension set of design primitives in the design field and establish the corresponding primitive knowledge base, rule base, instance library, etc.

Step 3: Search the primitive knowledge base, the rule base and the instance base for the features of the target primitives, find the design primitives that match the target primitives and analyze the matching degree based on the primitive extension inference model.

Step 4: For the design primitives that meet the requirements of the matching degree, the primitive extension reuse model is used to analyze the design reuse, and the primitives that meet the requirements of the reuse degree are optimized by extension, used as the design reuse object to be applied to the subsequent product design.

Step 5: Perform in-domain design extension transformation on the reused design primitives, obtain new design knowledge and classify them to form a new extension set of design primitives.

Step 6: Design reuse knowledge storage to prepare for the next product rapid configuration extension design.

4. Application Example

This paper takes the design of the turbine selection scheme of a large hydropower station as an example to illustrate the implementation of the extension design mode for complex products. Due to the imperfection of the design theory of large-scale hydraulic turbines and the complexity of the fluid movement inside the turbine, as well as the production mode of single-piece, small-batch and large-scale production of turbine product design, the design process of large-scale hydraulic turbine products has multi-level, multi-attribute and multi-level features. Constrained by multi-objective design features, the realization of the design scheme has become very cumbersome. Therefore, based on the extension design mode of complex products, this paper uses the methods of extension configuration, extension reasoning and extension reuse to realize a large-scale turbine selection scheme that meets the design constraints and technical requirements of the design head, the head section and the output. The rapid configuration of the design thus obtains the preliminary selection design scheme of the large-scale water turbine—that is, the extension design scheme. Through the extension demand analysis of Equations (6)–(11), the corresponding design demand interval of the large-scale turbine scheme design can be obtained. Table 1 gives the design demand parameter values of some runner types.

Table 1. Design requirement parameters of some runner models.

Feature Name	Eigenvalue	Feature Name	Eigenvalue
Inlet diameter of volute	6.200–7.800 (m)	Distribution circle diameter of guide vane	5.300–6.650 (m)
Inlet thickness of volute	35.000–65.000 (mm)	Height of guide vane	1.000–1.560 (m)
End thickness of volute	18.500–26.000 (mm)	Relative height of guide vane	0.200–0.275 (m)
Weight of volute	140.000–180.000 (T)	Weight of guide vane	1.650–2.820 (T)
Lining thickness of draft pipe	Long-term stability	Lining weight of draft pipe	15.700–21.100 (T)

Based on Equations (1)–(5), the volute target primitive $\boldsymbol{J}_{0-WK}$, the draft tube target primitive $\boldsymbol{J}_{0-WSG}$ and the guide vane target primitive $\boldsymbol{J}_{0-DY}$ are constructed, and the composite element $\boldsymbol{J}_{0-F}$ with "$\wedge$" and connection relationship between the three is formed, which is

$$J_{0-WK} = \begin{bmatrix} \text{volute} & \text{Inlet diameter(m)} & ([6.200, 7.800], 0.250) \\ & \text{Inlet thickness(mm)} & ([35.00, 65.00], 0.250) \\ & \text{end thickness(mm)} & ([18.50, 26.00], 0.250) \\ & \text{weight(T)} & ([140.00, 180.00], 0.250) \end{bmatrix}$$

$$J_{0-WSG} = \begin{bmatrix} \text{draft pipe} & \text{lining thickness(mm)} & ([15.000, 18.500], 0.500) \\ & \text{lining weight(T)} & ([15.700, 21.100], 0.500) \end{bmatrix}$$

$$J_{0-DY} = \begin{bmatrix} \text{guide vane} & \text{Inlet diameter(m)} & ([5.300, 6.650], 0.250) \\ & \text{Inlet thickness(mm)} & ([1.000, 1.560], 0.250) \\ & \text{end thickness(mm)} & ([0.200, 0.275], 0.250) \\ & \text{weight(T)} & ([1.650, 2.820], 0.250) \end{bmatrix}$$

$$J_{0-F} = \begin{bmatrix} \Gamma(\text{F}) & (\wedge)\text{volute features} & (J_{0-WK}, 0.346) \\ & (\wedge)\text{draft pipe features} & (J_{0-WSG}, 0.321) \\ & (\wedge)\text{guide vane features} & (J_{0-DY}, 0.333) \end{bmatrix}$$

Since most of the previous turbine design materials are recorded in paper form, some design data are often lost or forgotten during data collection or input, which will lead to a lack of completeness of the machine data in the turbine engineering database and knowledge base, thus making it difficult to perform complex product design reasoning and design reuse. The extension configuration mode and extension adaptive design mode proposed in this paper will be able to solve such problems better. First, based on the extension configuration mode of product design, the optimal extension reuse object at the whole machine level or the sub-set level is obtained, and we then analyze its adaptive design based on the product extension adaptive design mode, and finally obtain the optimal design result.

Based on the design features, the design object primitives are searched in the Francis turbine primitive knowledge base. For convenience of expression, some of the searched design object primitives are given without affecting the discussion of the paper, i.e.,

$$J_{0-F-d01} = \begin{bmatrix} \text{D01}-\text{GHY} & (\wedge)\text{volute features} & \left(\begin{bmatrix} \text{volute} & \text{Inlet diameter(m)} & (6.500, 0.250) \\ & \text{Inlet thickness(mm)} & (55.000, 0.250) \\ & \text{end thickness(mm)} & (20.000, 0.250) \\ & \text{weight(T)} & (193.383, 0.250) \end{bmatrix}, 0.346 \right) \\ & (\wedge)\text{draft pipe features} & \left(\begin{bmatrix} \text{draft pipe} & \text{lining thickness(mm)} & (18.000, 0.500) \\ & \text{lining weight(T)} & (15.742, 0.500) \end{bmatrix}, 0.321 \right) \\ & (\wedge)\text{guide vane features} & \left(\begin{bmatrix} \text{guide vane} & \text{Inlet diameter(m)} & (6.634, 0.250) \\ & \text{Inlet thickness(mm)} & (1.548, 0.250) \\ & \text{end thickness(mm)} & (0.233, 0.250) \\ & \text{weight(T)} & (2.138, 0.250) \end{bmatrix}, 0.333 \right) \end{bmatrix}$$

$$J_{0-F-d02} = \begin{bmatrix} \mathrm{D02-BMT} & (\wedge)\text{volute features} & \left(\begin{bmatrix} \text{volute} & \text{Inlet diameter(m)} & (7.500, 0.250) \\ & \text{Inlet thickness(mm)} & (40.000, 0.250) \\ & \text{end thickness(mm)} & (24.000, 0.250) \\ & \text{weight(T)} & (174.250, 0.250) \end{bmatrix}, 0.346 \right) \\ & (\wedge)\text{draft pipe features} & \left(\begin{bmatrix} \text{draft pipe} & \text{lining thickness(mm)} & (16.000, 0.500) \\ & \text{lining weight(T)} & (17.885, 0.500) \end{bmatrix}, 0.321 \right) \\ & (\wedge)\text{guide vane features} & \left(\begin{bmatrix} \text{guide vane} & \text{Inlet diameter(m)} & (5.400, 0.250) \\ & \text{Inlet thickness(mm)} & (1.100, 0.250) \\ & \text{end thickness(mm)} & (0.204, 0.250) \\ & \text{weight(T)} & (1.800, 0.250) \end{bmatrix}, 0.333 \right) \end{bmatrix}$$

$$J_{0-F-d03} = \begin{bmatrix} \mathrm{D03-LJX} & (\wedge)\text{volute features} & \left(\begin{bmatrix} \text{volute} & \text{Inlet diameter(m)} & (6.635, 0.250) \\ & \text{Inlet thickness(mm)} & (40.000, 0.250) \\ & \text{end thickness(mm)} & (24.000, 0.250) \\ & \text{weight(T)} & (154.000, 0.250) \end{bmatrix}, 0.346 \right) \\ & (\wedge)\text{draft pipe features} & \left(\begin{bmatrix} \text{draft pipe} & \text{lining thickness(mm)} & (16.000, 0.500) \\ & \text{lining weight(T)} & (21.085, 0.500) \end{bmatrix}, 0.321 \right) \\ & (\wedge)\text{guide vane features} & \left(\begin{bmatrix} \text{guide vane} & \text{Inlet diameter(m)} & (6.400, 0.250) \\ & \text{Inlet thickness(mm)} & (1.375, 0.250) \\ & \text{end thickness(mm)} & (0.215, 0.250) \\ & \text{weight(T)} & (2.100, 0.250) \end{bmatrix}, 0.333 \right) \end{bmatrix}$$

$$J_{0-F-d04} = \begin{bmatrix} \mathrm{D04-TSQ} & (\wedge)\text{volute features} & \left(\begin{bmatrix} \text{volute} & \text{Inlet diameter(m)} & (7.000, 0.250) \\ & \text{Inlet thickness(mm)} & (60.000, 0.250) \\ & \text{end thickness(mm)} & (24.000, 0.250) \\ & \text{weight(T)} & (215.000, 0.250) \end{bmatrix}, 0.346 \right) \\ & (\wedge)\text{draft pipe features} & \left(\begin{bmatrix} \text{draft pipe} & \text{lining thickness(mm)} & (16.000, 0.500) \\ & \text{lining weight(T)} & (17.885, 0.500) \end{bmatrix}, 0.321 \right) \\ & (\wedge)\text{guide vane features} & \left(\begin{bmatrix} \text{guide vane} & \text{Inlet diameter(m)} & (6.570, 0.250) \\ & \text{Inlet thickness(mm)} & (1.525, 0.250) \\ & \text{end thickness(mm)} & (0.2640.250) \\ & \text{weight(T)} & (2.772, 0.250) \end{bmatrix}, 0.333 \right) \end{bmatrix}$$

If there is a situation where the value of some design attributes is NULL, it needs to be ignored in the process of extension inference—that is, it is not used as a matching feature when matching primitive features, to ensure that the matching degree of primitive features by adjusting the consistency and reliability of primitive matching is improved. At the same time, combined with knowledge in the field of turbine design and the opinions of design experts in the field of turbine design, the diameter of the inlet section of the volute and the thickness of the end of the volute are taken as the forward design features, and the rest of the features are taken as the reverse design features.

Based on Equations (12)–(15), since there is no feature value of NULL, and the primitive features are completely matched, the matching degree sequence of primitives between each design object primitive J_{0-F-d} and the target primitive J_{0-F} is obtained:

$$K_{\varphi}(J_{0-F-d}) = (1.000, 1.000, 1.000, 1.000)$$

Based on Equations (16)–(23), the primitive feature similarity matrix of each design object primitive J_{0-F-d} and the target primitive J_{0-F} is obtained:

$$K_\sigma(J_{0-F-d}) = \begin{bmatrix} \begin{matrix}0.231\\0.400\\0.273\\-0.382\end{matrix} & \begin{matrix}1.000\\1.000\\1.000\\0.221\end{matrix} & \begin{matrix}0.335\\1.000\\1.000\\1.000\end{matrix} & \begin{matrix}0.616\\0.200\\1.000\\-1.000\end{matrix} \\ \begin{matrix}0.200\\1.000\end{matrix} & \begin{matrix}1.000\\0.583\end{matrix} & \begin{matrix}1.000\\0.003\end{matrix} & \begin{matrix}1.000\\0.583\end{matrix} \\ \begin{matrix}0.013\\0.026\\0.592\\0.669\end{matrix} & \begin{matrix}1.000\\1.000\\1.000\\1.000\end{matrix} & \begin{matrix}0.200\\0.402\\0.845\\0.706\end{matrix} & \begin{matrix}0.064\\0.076\\0.155\\0.047\end{matrix} \end{bmatrix}$$

Based on the discussion in Section 2.3—that is, using Equations (24)–(31)—the partial-level extension reuse matrix of each design object primitive J_{0-F-d} and target primitive J_{0-F} is obtained:

$$D(J_{0-F-d}) = \begin{bmatrix} \begin{matrix}0.131\\0.600\\0.325\end{matrix} & \begin{matrix}0.805\\0.792\\1.000\end{matrix} & \begin{matrix}0.834\\0.502\\0.538\end{matrix} & \begin{matrix}0.204\\0.792\\0.086\end{matrix} \end{bmatrix}$$

The whole-machine-level extension reuse sequence of each design object primitive J_{0-F-d} and target primitive J_{0-F} is:

$$D_F(J_{0-F-d}) = (0.346, 0.866, 0.629, 0.353)$$

Specifically, $\max(D_F(J_{0-F-d})) = D_F(J_{0-F-d02})$.

From the extension reuse sequence at the whole machine level, it can be seen that the best extension reuse object at the product level is $J_{0-F-d02}$; from the sub-level extension reuse sequence, it can be seen that the optimal extension reuse object of the sub-level level has changed—that is, the best extension reuse object of the volute is $J_{WK}|J_{0-F-d03}$, the optimal extension reuse object of draft tube is $J_{WSG}|J_{0-F-d02}$, the optimal extension reuse object of the guide vane is $J_{DY}|J_{0-F-d02}$, and the optimal extension reuse object $J_{WK}|J_{0-F-d03}$ of volute is not included in the optimal extension reuse object $J_{0-F-d02}$ of the whole machine level. Therefore, in the process of the adaptive design of a large-scale turbine selection scheme, it can be expanded from two aspects—that is, on the one hand, adaptive design can be carried out based on the best extension reuse object $J_{0-F-d02}$ at the product level; on the other hand, it can be based on the best extension of the nested volute and reuse the object $J_{WK}|J_{0-F-d03}$ for adaptive design. Regarding the relevant content of the extension adaptive design, this will be discussed in a future article.

5. Discussion

Based on the above theoretical content and application cases, it can be concluded that the method proposed in this paper has a strong theoretical foundation. From the aspects of knowledge extension modeling, knowledge extension transformation, knowledge extension reasoning, knowledge extension reuse and extension pattern generation, an extension configuration design method system for complex product scheme design is formed, which has good engineering applicability.

This method can generate product design knowledge through local configuration design features by establishing a knowledge primitive model of complex product configuration design and adopting a consistent configuration design problem description

combining quantitative and qualitative methods. This method carries out the extension transformation analysis of a complex product scheme design from the perspective of the integration of the design domain, design features and eigenvalues; realizes the rapid design of complex products through the rapid configuration of an extension reasoning model and extension reuse model of a complex product scheme design; and adopts the extension logic reasoning for deep-level knowledge processing to analyze the convergence of the configuration variables of the complex product scheme design and solve the problem of configuration contradiction. At the same time, the basic steps and algorithm of the extension design pattern of complex product scheme design and rapid configuration design are analyzed, and the application of an example also verifies the effectiveness of the algorithm.

In addition, applying knowledge extension reuse technology to complex product scheme design not only makes product design standardized and systematic, but also expands the application field of expert systems, provides a theoretical basis for computer-aided product conceptual design and plays an important role in the smooth implementation of complex product design scheme development. However, the analysis of the application ability of the relevant extension design model in this paper is not extensive enough—that is, the retrieval efficiency of the model, the extension reuse ability, etc., are neglected. Thus, this content will be included in future work.

6. Conclusions

Aiming at the multi-level, multi-attribute and creative product structure configuration process of large-scale complex products, this paper studies the extension design mode of the rapid configuration of large-scale complex product conceptual design, and puts forward a rapid configuration design model of large-scale complex product conceptual design. The specific results and conclusions are as follows. (1) For static design information, behavioral design information and relational design information in conceptual design, the matter-element model, event-element model and relational element model are established, respectively, to realize the formal modeling of design information. (2) An extension transformation model for rapid configuration design is presented, which provides support for generating richer knowledge of rapid configuration design. (3) An extension reasoning model for rapid configuration design is established, which provides support for improving the reasoning ability of product rapid configuration design. (4) An extension reuse model for rapid configuration design is proposed, which provides support for enhancing the knowledge reuse ability of product rapid configuration design, and realizes the rapid configuration of complex product conceptual design.

Author Contributions: Conceptualization, T.W.; Data curation, T.W., H.L. and X.W.; Formal analysis, H.L. and X.W.; Funding acquisition, T.W.; Methodology, T.W.; Project administration, T.W.; Writing—original draft, T.W.; Writing—review & editing, T.W. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by Natural Science Foundation of Jiangsu Province of China (No. BK20221481), the National Natural Science Foundation of China (No. 51775272, No. 51005114).

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The labeled dataset used to support the findings of this study are available from the corresponding author upon request.

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

References

1. Liu, J.; Zhang, Z.; Evans, R.; Xie, Y. Web services-based knowledge sharing, reuse and integration in the design evaluation of mechanical systems. *Robot. Comput.-Integr. Manuf.* **2019**, *57*, 271–281. [CrossRef]
2. Wan, S.; Li, D.; Gao, J.; Roy, R.; He, F. Maintenance and service knowledge reusing for CNC machine tools. *Comput. Integr. Manuf. Syst.* **2018**, *24*, 126–141.

3. Mourtzis, D.; Doukas, M.; Giannoulis, C. An Inference-based Knowledge Reuse Framework for Historical Product and Production Information Retrieval. *Procedia CIRP* **2016**, *41*, 472–477. [CrossRef]
4. Mourtzis, D.; Doukas, M.; Giannoulis, C. Interface model-based configuration design of mechatronic systems for industrial manufacturing applications. *Robot. Comput. Integr. Manuf.* **2019**, *59*, 373–384.
5. Liu, F.; Zhang, Y.; Zheng, C.; Qin, X.; Eynard, B. Survey of Configuration Design Approaches: A Focus on Design of Complex Industrial Manufacturing Systems. *Procedia CIRP* **2019**, *81*, 340–345. [CrossRef]
6. Badurdeen, F.; Aydin, R.; Brown, A. A multiple lifecycle-based approach to sustainable product configuration design. *J. Clean. Prod.* **2018**, *200*, 756–769. [CrossRef]
7. Aydin, R.; Brown, A.; Badurdeen, F.; Li, W.; Rouch, K.E.; Jawahir, I.S. Quantifying impacts of product return uncertainty on economic and environmental performances of product configuration design. *J. Manuf. Syst.* **2018**, *48*, 3–11. [CrossRef]
8. Zhang, L.; Zhang, S.; Liu, X.; Wang, Y.; Zhang, P. Reconfiguration technology of configuration design for complex product based on GRA and weighted sequence cross. *Comput. Integr. Manuf. Syst.* **2015**, *21*, 2564–2576.
9. Xu, B.; Yang, Y.; Li, J. Fast configuration model and algorithms for large-scale container crane product. *Comput. Integr. Manuf. Syst.* **2015**, *21*, 1177–1186.
10. Ren, S.; Zhao, Y.; Hong, H. A Retrieval Method for Similar Cases with Multiple Attributes in Low-carbon Design. *J. Mech. Eng.* **2019**, *55*, 161–171. [CrossRef]
11. Zhao, Y.; Zhou, J.; Hong, H. Overview and prospects of extension design methodology. *Comput. Integr. Manuf. Syst.* **2015**, *21*, 1158–1167.
12. Cai, W.; Yang, C. Basic theory and methodology on Extenics. *Chin. Sci. Bull.* **2013**, *58*, 1190–1199.
13. Ren, J. Technology selection for ballast water treatment by multi-stakeholders: A multi-attribute decision analysis approach based on the combined weights and extension theory. *Chemosphere* **2018**, *191*, 747–760. [CrossRef] [PubMed]
14. Zhou, Y.; Shi, J.; Wu, L. Application of Extension Theory in Emotion Management. *Procedia Comput. Sci.* **2017**, *122*, 502–509. [CrossRef]
15. Ma, L.; Chen, H.; Yan, H.; Li, W.; Zhang, J.; Zhang, W. Post evaluation of distributed energy generation combining the attribute hierarchical model and matter-element extension theory. *J. Clean. Prod.* **2018**, *184*, 503–510. [CrossRef]
16. Wang, M.H.; Tseng, Y.F.; Chen, H.C.; Chao, K.H. A novel clustering algorithm based on the extension theory and genetic algorithm. *Expert Syst. Appl.* **2010**, *46*, 146–154. [CrossRef]
17. Zhao, Y.; Zhang, M.; Su, N.; Chen, J. Product Rapid Configuration Design Based on the Extension Classification Analysis Model. *J. Southwest Univ.* **2014**, *36*, 201–208.
18. Wu, P.; Liu, J. Solution to Creating Multi-function Product Based on Extension Design Methods. *J. Guangdong Univ. Technol.* **2015**, *32*, 10–17.
19. Wang, T.C.; Yang, A.J.; Zhong, S.S.; Zhao, Z.D. Extension adaptive design model of scheme design for complex mechanical products. *Tehnički Vjesnik/Technical Gazette* **2014**, *21*, 123–133.
20. Wang, T.; Xie, Y.; Yan, H. The Extension Adaptive Design Model for Mechanical Product Lifecycle. *Rev. Comput. Eng. Stud.* **2015**, *2*, 7–13. [CrossRef]
21. Wang, T.C.; Hu, X.X.; Zhong, S.S.; Zhang, Y.J. Research on Extension Knowledge Base System for Scheme Design of Mechanical Product. *Math. Model. Eng. Probl.* **2016**, *3*, 141–145. [CrossRef]
22. Wang, T.; Wang, J. A fault diagnosis model based on weighted extension neural network for turbo-generator sets on small samples with noise. *Chin. J. Aeronaut.* **2020**, *33*, 2757–2769. [CrossRef]
23. Wang, T.C.; Tong, C.S.; Yao, S.H. An Improved Prediction Algorithm of Seamless Tubing Corrosion Based on Extension Neural Network. *Concurr. Comput. Pract. Exp.* **2019**, *31*, e4801. [CrossRef]