International Journal of Grid and Distributed Computing
Vol. 9, No. 4 (2016), pp.349-360
http://dx.doi.org/10.14257/ijgdc.2016.9.4.31

Process Backtracking and Reconstruction based on Task Chain Model

Jing Chen [a,b*], Dewen Seng [a] and Xujian Fang [a]

[a]*School of Computer Science and Technology, Hangzhou Dianzi University Key Laboratory of Complex Systems Modeling and Simulation, Ministry of Education, China*
[b]*Yangtze Delta Region Applied Mathematics Institute of Zhejiang*
cj@hdu.edu.cn

Abstract

Fundamental information of the design unit is described using the correlation between the nodes in the task chain. Data transfer between design units is formulated as fundamental data transfer, design rule transfer and path scheme transfer, respectively. The design process is stored with the node as the unit by using the algorithm for decomposing correlated nodes. The reconstruction method is employed to eliminate the redundant nodes that exist in various previous design processes, alleviating the degree of coupling. The performance of the proposed scheme is verified by applying it to the development of the low-voltage appliance.

Keywords*: Design node, Design process, Process reconstruction, Task chain model*

1. Introduction

Reuse in the design process refers to the use of existing design knowledge, history and cases again to improve design efficiency. Most of existing designs are variants of the previous designs. Fully exploiting existing designs for reuse can greatly improve the efficiency and quality of design. The design process which accounts for 5% of the product's total budget, determines approximately 70％-85％ of the products quality. Paying close attention to the design process can substantially reduce the product's cost, and improve the product's manufacturability and maintainability. Therefore, efficient design provides a reliable guarantee for high product quality, low product cost, and shortened development cycle of the product. Due to studies conducted home and abroad, some representative theories and methods have been proposed for acquisition and evaluation of design steps [1], reuse of manufacturing knowledge in the design process [2,3], and for the representation and reuse of the formal knowledge [4, 5]. Literature [6] presents a new interactive method for adaptation knowledge elicitation, acquisition and reuse, proposes the integration of the Constraint Satisfaction Problem approach into the Case Based Reasoning process to improve the case representation and the adaptation of past experiences. A new reusable template to design reuse design processes was presented [7], which include four important characteristics: reusability, hierarchical decomposition and composition, domain independence, and decision-centric. Literature [8] presents a process template includes process modeling level and project running level for design resource integration. Literature [9] put forward a concept of design process gene to explain the principle and procedures of design process reuse. Literature [10] proposed a feedback model based on the tentative design chain. The methods of classified consumer intention and concept binding are used to describe the intuitionist instance into a disciplinary design chain.

ISSN: 2005-4262 IJGDC

Works about reuse of the product's design knowledge are mainly focused on the representation, acquisition, organization, and indexing of the design knowledge [11]. The instances most similar to the problem at issue are selected based on constraints constructed from distinctive features. But the instances in the design system are usually limited to the instance-specific knowledge, and the universal knowledge cannot be used to dynamically generate the information about the process of product design. Reuse-based design is by no means the duplication of the process of product design. Instead, it applies similar design processes to other correlated design tasks, including design constraints, correlation and design experience. During the reuse, the design scheme that matches the design objective may have more than one solution at the same time. Reuse can be done by recording the path of the entire design task, but different design paths have many identical or correlated design nodes, which are repeatedly recorded and utilized during different design processes. Meanwhile, in the methods for recording design paths, existing design paths can hardly be used to design similar products of different types. With increase in the amount of instance data, the historical paths contain so much repeatedly recorded redundant information, directly affecting matching in the reuse process and the efficiency of design.

To address the problems above, the design process is formulated in this paper as the correlation between the nodes in the design task. The incidence matrix stores all possible correlations between design nodes, including the serial, parallel and coupled nodes needed to constitute a complete task chain. Meanwhile, it contains multiple matched design paths to solve a target task. In these design paths, there may exist conflicting relational nodes. The correlation between nodes is stored using the algorithm for decomposing the incidence matrix. Optimization and reuse of nodes throughout the design network is done using the combination algorithm. The process backtracking and reconstruction based on the task chain model does not refer to the storage of a certain design process. Instead, it aims to utilize the input, output and coupling relations between the nodes of the entire design network at issue, in order to reduce redundancy of design nodes, and enhance effectiveness and efficiency of reconstruction.

2. Node Definition and Representation

By constructing the design chain, the knowledge regarding the design of complicated products can provide a universal model for knowledge acquisition and inference. The model consists of the design nodes and the arcs. The design node is a design unit for solving the problem. The user acquires design information from design nodes and determines the hierarchy of the design model using correlations among nodes. The design nodes are categorized into the target nodes and the normal nodes. The target node can match with the design objective and obtain the first-layer correlated nodes of the incidence matrix. It has the descriptive properties of normal nodes and can also be used as the node during the design process. Each element denotes a node in the task chain. It is represented as a five-element set, as shown in Table 1.

$$Nd = \{Bs, Rl, Pl, Ip, Cf\} \quad (1)$$

where Bs denotes the set of basic node attributes, R1 denotes the set of correlated nodes, P_1 denotes the set of correlation attributes, *Tf* denotes the set of transferring attributes, *Cf* denotes the set of attributes for judging node conflicts. Major events that occur at each phase of product design and the indexes of each phase are stored into the data structure defined by this model. Basic attributes of the node include basic information about the node, such as design parameters, expressions, retrieval codes, and notes. The set of correlated nodes denotes all nodes correlated to the current node, *i.e.* $\{A_1,B_1,\ldots\}$. The set of correlation attributes corresponds to the set of all nodes correlated to the current node, representing the set of child nodes correlated to the current node, *i.e.* $\{A_2, B_2,\ldots\}$,$A_2=\{a,$

b}, B_2={*c*, *d*, *e*}, where the children a and b combine to yield the node A_2, and the children c, d and e combine to yield the node B_2. The transferring attribute refers to the output manner of the current node, which can be classified into the basic data, design rule and path scheme. The attributes for judging node conflicts relate to node effectiveness, consistency and uniqueness.

Table 1. Representation of Node Attributes

Attributes	Descriptions
Basic attributes, *Bs*	Analysis of real-time variables input and output for basic data and the CAE system, design parameters, expressions, retrieval codes, notes and other basic information regarding the node
Correlated nodes, Rl	Set of all nodes correlated to the current node
Combination attributes, *Pl*	Representation of node relations parallel to serial input, serial output and coupling
Transferring attributes, *Ip*	Data transfer relation between two neighboring nodes in the task chain structure including basic data transfer, design rule transfer and path scheme transfer
Conflict judgment, *Cf*	Return the judgment on node effectiveness, consistency and uniqueness

Description of the node's basic information and correlated attributes not only achieves a balance between adequacy of knowledge representation and effectiveness of knowledge inference but also indicates the inheritance relation between knowledge correlation and dynamic information. It forms the foundation for virtual manufacturing-oriented knowledge correlation in the design process.

3. Correlation Constraints between Nodes in the Task Chain

3.1. Representation of Nodes in the Task Chain

The design process may be influenced by different operational environments and model parameters. Thus, every variation of the design parameters has its impact on the entire design process. The purpose of obtaining the target nodes is to determine the correlated nodes in the design process subject to design constraints on the workpieces' types, materials, sizes and batch quantities. In addition to the basic information of the nodes in the task chain, the correlation between nodes also needs to be described.

The design information is constantly transferred between design activities during the process of product design. The directed graph is used to represent the design process. The directed graph is a set of nodes and the directed arcs linking the nodes, denoted by $G=<V,E>$, where $V=\{v_1,v_2,\ldots,v_n\}$ is the set of nodes, representing n design activities; $E=\{e_1,e_2,\ldots,em\}$is the set of m arcs, representing the path and direction of information transfer. Each element in E corresponds to two nodes in *V*.

The task chain consists of task nodes, and a task node represents a basic element of design. When the transferring relation exists between two neighboring nodes in the task chain, these two nodes are considered correlated. The data transferring relation between two neighboring nodes in the task chain structure can be categorized into basic data transfer, design rule transfer and path scheme transfer. Basic parameter transfer refers to the transfer of the designed product's resource information, design object, design type, design function and data table. The transfer of parameters in the design formula involves the data of the design manuals and the design variables obtained in real time. It includes the transfer of empirical parameters and deterministic parameters. The design rule transfer means the transfer of the rules on the current node's design parameter constraint, performance constraint, function constraint, design structure constraint and conflict

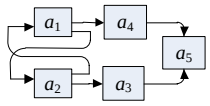
constraint during the design process. The transfer of the extracted and transformed design information incorporates regular description of the solving process. The path scheme transfer is a general procedure for solving problems. It can learn from similar task chains that have existed before, *i.e*. remembering how it solve the problem itself and applying the experience to other problems.

3.2. Description of Correlation between Nodes in the Task Chain

The incidence matrix stores correlations between all task nodes. These nodes may form more than one task chain. A complete task chain corresponds to a structural matrix of design. The incidence matrix of design is constructed based on the transferring relation between nodes. The element a_{ij} of the matrix is defined as follows:

$$a_{ij} = \begin{cases} 0 & a_i \ is \ not \ the \ input \ node \\ m & a_i \ is \ the \ input \ node \end{cases} \quad (2)$$

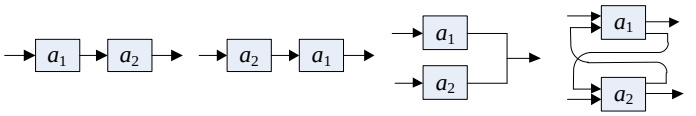
The leading diagonal element of the matrix a_{ii}=0. If $a_j \rightarrow a_i$, it means that the node a_j can provide input for the node a_i. The value of the input $m \in \{1,2,3\}$,meaning that the transferring relation between nodes is the transfer of basic data, the transfer of expertise, and the transfer of path scheme, respectively. Otherwise, a_{ij}=0.



	a_1	a_2	a_3	a_4	a_5
a_1		1		1	
a_2	1		1		1
a_3					1
a_4					1
a_5					

Figure 1. Design Correlation Matrix

The incidence matrix of design stores the information dependency relationship between constitutional activities and activities. As shown in Figure 1, non-zero elements in each row represent the input information needed to carry out this activity. And the non-zero elements in each column represent the activities which need the output of this activity. From the perspective of information flow, the relation between activities of product design can be categorized into serial input, serial output and coupling. Figure 2(a) means that the node a_2 depends on the node a_1, Figure 2(b) means that the node a1 depends on the node a_2, Figure 2(c) means that the relation between a1 and a_2 is parallel, and Figure 2(d) means the nodes a_1 and a_2 are mutually dependent.



	a_1	a_2
a_1		0
a_2	m	

(a) Serial input

	a_1	a_2
a_1		m
a_2	0	

(b) Serial output

	a_1	a_2
a_1		0
a_2	0	

(c) Parallel

	a_1	a_2
a_1		m
a_2	m	

(d) Coupling

Figure 2. Activity Relationship of Design Node

4. Backtracking and Reconstruction of Nodes in the Task Chain

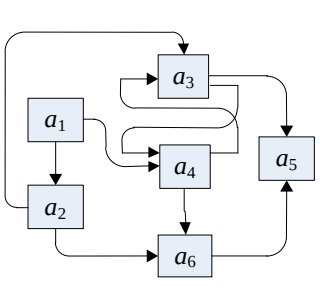
Figure 3 shows that as the target node of all task chains, the node a_5 is located in a row where the elements are zeroes in the structural matrix of design, *i.e*. the node a_5 does not provide any output. As the starting node of all task chains, the node a_6 is located in a column where the elements are zeroes in the structural matrix of design, meaning that other nodes do not provide input fora6 and that a_6 only contains the output information. Based on the input and output characteristics acquired from the incidence matrix, the decomposition method can be used to obtain the process nodes correlated to the target node and the starting node. The following rules should be followed: (1) If the elements in a row of the incidence matrix are all equal to zero, it means that carrying out this task yields no output, *i.e*. this node is a target node; (2) if the elements in a column of the incidence matrix are all zeroes, it means that carrying out this task requires no input, *i.e*. this node is a starting node; (3) in the incidence matrix of the task node, the nodes corresponding to all non-zero elements in the column of the current node are the left neighbors of the current node, *i.e*. the input of the current node comes from its left neighbors. Steps of the algorithm are given below:

Step 1. Consider that the incidence matrix has k correlated nodes. Let i =2. The nodes located where the elements of the row are all zeroes are defined as the target nodes, and the set of these nodes is denoted by O_1. The decomposition set is denoted by D_1, and $D_1 \cap \{P_1\}=\Phi$.

Step 2. If $D_i=\Phi$, then the algorithm ends. Otherwise, obtain each decomposition node Tijin the set O_i. Assign a number to the correlated node, which is equal to the number of zero-valued elements in the row of this correlated node. Let j denote the serial number of the node corresponding to non-zero elements in the column vector of all nodes, and $0 \le j < k$. Let $Ct(i,j)$ dente the number of correlated nodes in the set O_i.

Step 3. If the elements of the column of the decomposition node T_{ij} are all zeroes, then $i=i+1$ and jump to Step 2. Otherwise, if the column vector of the nodes in the set O_i has no zero element, then $j=j+1$ and repeat Step 3. If the set of nodes O_i is correlated to the end node of the jth decomposition node in the set of nodes O_i -1, then update the set O_i, replace the $Ct(i, j)$ serial-node links with the obtained $Ct(i, j) \times Ct(i\text{-}1, j)$ serial-node links. Here, the number of serial-node links in O_i is $Ct(i, j) + Ct(i, j\text{-}1)$. $j=j+1$ and repeat Step 3.

The number of the algorithm's iterations is defined as the sequence number of this node, and is used as the index for judging parallel input of the task nodes.



	a_1	a_2	a_3	a_4	a_5	a_6
a_1		1		1		
a_2			1			1
a_3				1	1	
a_4			1			1
a_5						
a_6					1	

Figure 3. Node Relationship of all Tasks Associated with the Target Node

The node where the number of iterations is larger than 1 between the same nodes is defined as the coupling node in the algorithm. The nodes which have the coupling relation are regarded as a whole. Meanwhile, because the design objective is clear, the target node cannot be the coupling node. Figure 4 shows the correlations of all task nodes by correlating with the target node a_5. From the incidence matrix, it can be seen that among

the elements where the node a_5 is located, the nodes correlated with a_5 are a_2, a_4 and a_7. Among the elements where a_2 is located, the node correlated with a_2 is a_1. Among the elements where a_4 is located, the node correlated with a4 is a6. Among the elements where a_7 is located, the node correlated with a_7 is a_6. Among the elements where the node a_1 is located, the nodes correlated with a_1 are a_2, a_3, a_4 and a_6.The five serial-node relations in Figure 4 are acquired from running the decomposition algorithm.

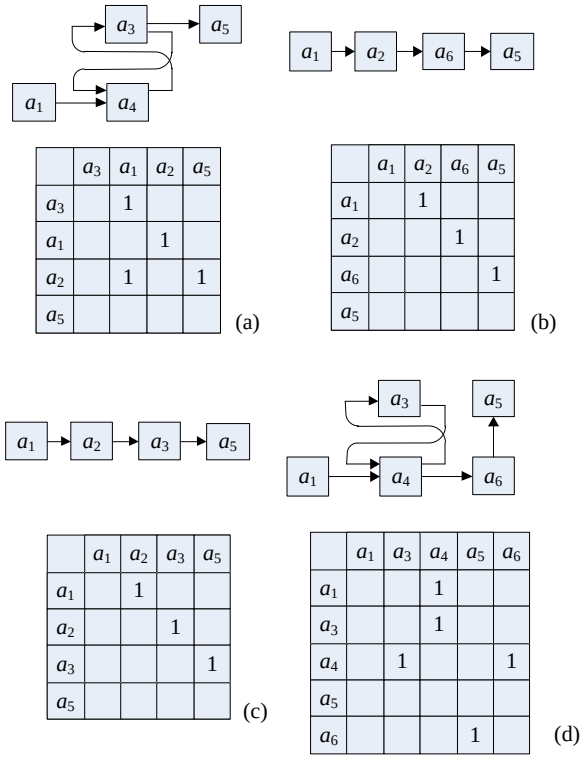


Figure 4. Relationship of Serial Node

What is acquired from the decomposition algorithm is the serial relation solution that consists of nodes correlated with both the target node and the starting node in the incidence matrix. Based on this, the parallel relation between nodes can be obtained from the parallel attributes of nodes. Reconstruction of parallel nodes means that all serial nodes in the incidence matrix are merged in an ordering manner based on parallel attributes of nodes. Steps of the method are described as follows:

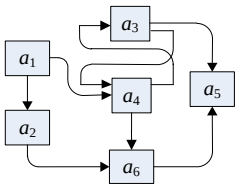
Step 1: The numbers of iterations of the decomposition nodes acquired from the decomposition algorithm are sorted out in a reverse order. Define it as the sort index of the node, and $0 \le i \le n$. Determine the parallel attributes of the task nodes using the index nodes. Set the recombinant index $i = 2$.

Step 2: Obtain the combination type j of parallel nodes, according to the parallel attributes of the node N_i, and $0 \le j \le m$.

Step 3: If $j > m$, jump to Step5. Otherwise, determine whether the jth parallel attribute $Ct(i,j)$ of the node i exists in the current correlated node. If the left neighbor of the current node exists, then jump to Step 4. Otherwise, $j = j + 1$, and repeat Step3.

Step 4: Merge parallel nodes of $Ct(i,j)$. Regard all merged nodes as a set of nodes, delete the serial-node link that has been merged, and let $j = j + 1$.

Step 5: If $i > n$, the algorithm ends; otherwise, $i = i + 1$, and jump to Step2.



	a_1	a_2	a_3	a_4	a_5	a_6
a_1		1		1		
a_2						1
a_3				1	1	
a_4			1			1
a_5						
a_6					1	

Figure 5. Merge Task Chain Nodes

The node a_1 is used as the merging node based on the ordering of indexes in Figure 5. The node attributes include the parallel attribute C_{11}={a_3:a_6}.Judging from the attribute, if the right neighbor of a_1 is a_3 or a_6, then the two nodes are merged as the parallel node. Figure 5 shows the task link acquired by merging nodes in Figures 4(a) and 4(e).

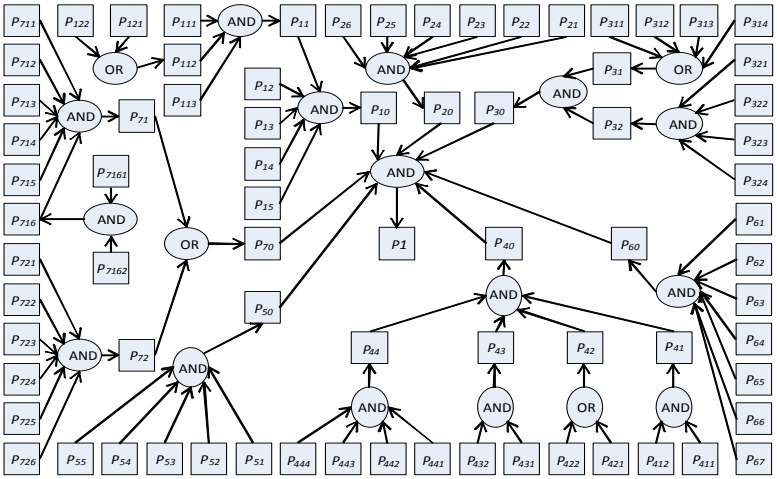
The process with the shortest path is chosen to be designed actually by distinguishing between the centralized and distributed principles based on the path length (*i.e*. the number of task nodes during design). The degree of coupling is obtained from the structural matrix of design during the design process. The similarity between two vectors is computed using the correlation between task chains after the task chain vectors and the query vectors are determined. For the query vector q with n attributes and the task chain vector q, their similarity can be computed as follows:

$$similarity \quad (q, d) = \frac{\sum_{i=1}^{n} q_i d_i}{\sqrt{\sum_{i=1}^{n} (q_i)^2 \sum_{i=1}^{n} (d_i)^2}} \tag{3}$$

After the system responds to the knowledge-selected application request, the correlated chain mapped to the target function is called to obtain the independent information of nodes in a task through analysis, including node name, node parameter, type of judgment, input/output, execution index and type of transferred value.

5. Case Study

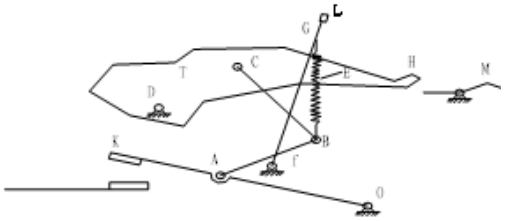
The design of low voltage circuit breaker is an interdisciplinary process that involves dynamics, statics, electromagnetic, *et al*. Parameters of different low voltage circuit breakers need to be selected during reuse-based design, especially for the processing of coupling parameters and the setting of empirical nodes. Consider the design of breaker. Figure 6 shows the correlations between task chain nodes of the designed breaker. The chain structure of the model is obtained in this way. The incidence matrix of the breaker's nodes is constructed based on the structure of the task chain. The node library stores the information of nodes of previous breakers. Besides the nodes in the design manual, the empirical nodes that have been confirmed are also added.



P_{11} : Material requirements
P_{111}: Arc resistance
P_{112}: Conductive
P_{113}: Welding resistance
P_{12} : Opening distance
P_{13} : Contact pressure
P_{14} : Contact overtravel
P_{15} : Contact selection
P_{121}: Fixed welding resistance
P_{122}: Static welding resistance
P_{21} : Strips and contact distance
P_{22} : FIlms number
P_{722}: Primary winding turns
P_{23} :Gate and tablets distance
P_{24} :Arc guiding plate type
P_{25} : Plate and contact distance
P_{26} :Arc guiding plate angle
P_{31} :Transmission selection
P_{311}:Lever
P_{312}:Lever transmission
P_{313}:Solenoid drive
P_{314}:Gas hydraulic transmission
P_{32} :Actuator
P_{321}:Closing torque
P_{322}:Closing counter moment
P_{723}:Secondary winding turns
P_{323}:Open torque
P_{324}:Open counter moment
P_{41} :Thermal sensitivity
P_{411}:Physical properties
P_{412}:Combination of material
P_{43} :Stress analysis
P_{431}:Use stress
P_{432}:Limit stress
P_{44} :Material requirements
P_{441}:Temperature range
P_{442}:Heating type
P_{443}:Tensile properties
P_{724}:The total ampere-turns
P_{444}:Load stress
P_{51} :Allow volume
P_{52} :Cross-sectional area
P_{53} :Working air gap
P_{54} :Turns
P_{55} :Core material selection
P_{61} :Core size
P_{62} :Armature size
P_{63} :Working air gap
P_{64} :Voltage coil turns
P_{65} :Voltage coil diameter
P_{66} :Coil temperature
P_{725}: Secondary winding voltage
P_{67} :Core material
P_{71} :General TA
P_{711} :Setting current range
P_{712} :Primary winding turns
P_{713} :Turns ratio difference
P_{714} :Current ratio
P_{715} :Saturation current multiples
P_{716} :Core design
P_{7161}:Core structure type
P_{7162}: Magnetic induction
P_{72} : Saturation TA
P_{721} : Load range
P_{726} : Core cross-sectional area

Figure 6. Correlation of Nodes in the Breaker's Task Chain

Consider the operating mechanism of the low voltage breaker's contact head. The closing and opening actions of the breaker are carried out through the use of the actuating mechanism, the trip-free mechanism, main axle and the tripping shaft. Mechanism sketch of the trip-free breaker is shown in Figure 7.



L—handle, *T*—trip, *M*—lock, *K*—contact head, *CB*—upper connecting bar, *BA*—lower connecting bar, *E*—main spring

Figure 7. Mechanism of Circuit Breaker

When the breaker is switched off, trips free and is located in a re-tripping position, *OA*, *AB*, *BC* and the fixed point *CO* form the quadratic crank mechanism. When the mechanism is at the trip-free state, *OA*, *AB*, *BC*, *CD* and *DO* form the five-rod mechanism. The target task is to design the contact head's actuating mechanism by computing the switching torque and counter torque of the handle.

$$F_{AB} = \sin\alpha \times \mathrm{F}/\cos\alpha$$
$$M_h = F_{AB} \times L_{AB} \quad (4)$$
$$M_f = 3F_0 \times \mathrm{L_k}$$

Where F denotes the main spring's pulling force, α denotes the switching angle, θ denotes the opening angle, M_h denotes the switching torque of the contact system, M_f denotes the counter switching torque. It can be alternatively computed by calculating the opening torque ratio of the handle.

$$M_c = F \times \lambda$$
$$M_F = 3F_f \times \mathrm{L_k} \tag{5}$$

where F_c denotes the handle's opening force, M_c denotes the opening torque, M_F denotes the counter closing torque. It can also be computed by calculating the tripping force.

$$F_{BC} = F_{AB} \times \mathrm{Fcos}\alpha$$
$$F_H = F_{BC} \times \mathrm{L_{BC}}/\mathrm{L_H} \tag{6}$$

where F_{BC} denotes the component force of the connecting bar, F_{AB} denotes the component force of the lower connecting bar, F denotes the main spring's stretching force, α denotes the switching angle, F_H denotes the tripping force. The design tasks can be accomplished in many ways. And these computing methods share several same parameters. The process of obtaining a solution to the design task is the process of correlating with the right design paths according to the requirements on types and design nodes. The node is defined as the unit to store the task chains. The addition of design nodes is set to be constantly maintained during daily design. Description of the correlated task chains above is given below:

Consider the node task_1. The correlation parameters include the switching torque M_n, counter switching torque M_f, input variables x_{11} and x_{22}, and the output variable is the torque ratio y_{11}. The input variables of the switching torque M_h and the counter switching torque M_f are unknown. Thus, they are decomposed by the nodes 2 and 5, and represented by task_2 and task_5. The correlating parameters of task_2 are the component force of the lower connecting bar (denoted by F_{AB}), and the vertical distance between the component force and the pivot point (denoted by L_{AB}); the input variables are x_{31}, x_{32} and the output variable is y_{21}. They can be represented by the following incidence matrix as shown in Figure 8.

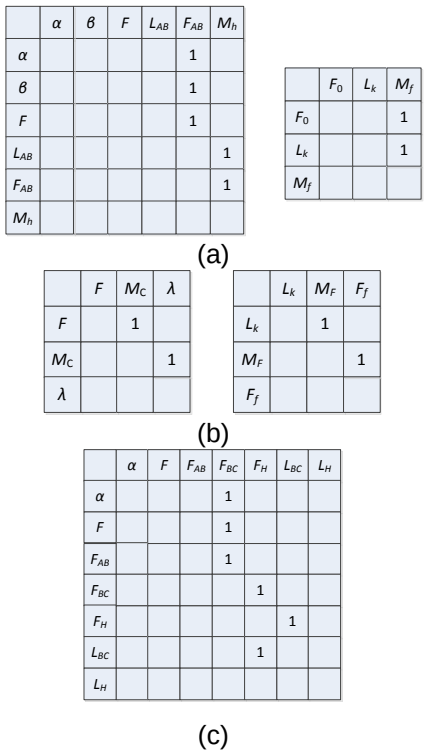
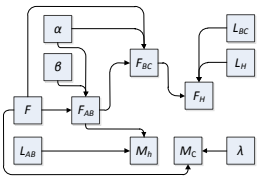
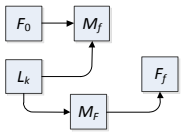


Figure 8. Incidence Matrix of the Contact Head

Note that the design process is represented by the task chain, and the parameter nodes that form the task chains are interrelated. Therefore, to reduce storing redundancy and represent information of the process efficiently, the breakers that have the same functions but are of different types are stored using the nodes. The correlation for the design of the breaker's contact head can be expressed as a complete incidence matrix as shown in Figure 9.



	α	β	F	L_{AB}	F_{AB}	F_{BC}	M_h	F_H	M_C	L_{BC}	L_H	λ
α		1			1	1						
β					1							
F					1	1						
L_{AB}							1					
F_{AB}							1					
F_{BC}												
M_h												
F_H						1						
M_C			1									
L_{BC}								1				
L_H								1				
λ									1			



	F_0	M_f	L_k	M_F	F_f
F_0		1			
M_f					
L_k		1		1	
M_F					1
F_f					

Figure 9. Chain Structure of the Task Chain Model for the Design of the Breaker's Contact Head

It can be judged from parameters (*i.e*. L_{AB}, L_{BC}, L_H, α, β, γ) that the design process varies with the types of the breakers. But most of the nodes are the same. During the actual process of node reconstruction, while meeting all constraints, the task chains are correlated with different design nodes based on parameter values. The reconstruction algorithm can compute the value of F_{BC} by obtaining several different task chain models. Its correlation is shown in Figure 10.

	α	β	F	F_{AB}	F_{BC}
α				1	
β				1	
F				1	
F_{AB}					1
F_{BC}					

	F_H	L_{BC}	L_H	F_{BC}
F_H				
L_{BC}	1			
L_H	1			
F_{BC}			1	

Figure 10. Correlation for Task Chain Reconstruction

6. Conclusions

This paper proposes a process reconstruction method based on the task chain model, and studies the knowledge correlating method for the design of products. The proposed method has the following characteristics. (1) It extends the knowledge from the static domain to the dynamic domain, and is focused on describing the correlation of nodes and the inheritance of the dynamic information. (2) The decomposition and reconstruction of the design tasks rely on the incidence matrix. It is not a simple combination of separate design processes. Instead, it applies similar design processes (*e.g*. design constraints, correlations, and design experience) to other correlated design tasks again. The task

chains are stored with the node as the unit, resulting in greatly reduced node redundancy and enhanced efficiency for design and storage.

Acknowledgments

This work is supported by the National Natural Science Foundation of China (No.61473108), the Natural Science Foundation of Zhejiang Province (No. LQ13F020005), and the research foundation of the Education Department of Zhejiang Province (No. Y201430884).

References

[1] Y. Ishino, Y. Jin. An information value based approach to design procedure capture. Advanced Engineering Informatics, 1, 20 (2006) ,pp.89-107.
[2] L. Krogstie, P. Andersson. A Case Study on Reuse of Manufacturing Knowledge - Comparing Defense Practices with Automotive & Aerospace Practices. Procedia Cirp, 3,(2012),pp.430–435.
[3] W. L. Mikos, J. C. E. Ferreira, P. E. A. Botura , L. S. Freitas. Development of a system for distributed sharing and reuse of design and manufacturing knowledge in the PFMEA domain using a description logics-based ontology. Automation Science and Engineering (CASE), 2010 IEEE Conference on. IEEE, (2010), pp. 598 - 603.
[4] J. H. Wu. A design methodology for form-based knowledge reuse and representation[J]. Information & Management, 7,46(2009),pp.365–375.
[5] D. Richards, P. Compton. Combining Formal Concept Analysis and Ripple Down Rules to Support the Reuse of Knowledge. Software Engineering and Knowledge Engineering(1997).
[6] E. R. Reyes, S. Negny, G. C. Robles, J. M. L. Lann. Improvement of online adaptation knowledge acquisition and reuse in case-based reasoning: Application to process engineering design. Engineering Applications of Artificial Intelligence, 41(2015) ,pp.1–16.
[7] J. Yu, C. Y. Lu. Design synthesis approach based on process decomposition to design reuse. Journal of Engineering Design, 7,23 (2012) ,pp.526-543.
[8] Y. Li, W. Zhao. Process template based flexible resource integration for product innovative reuse design. International Journal of Computer Integrated Manufacturing, 1, 24(2011) ,pp.53-73.
[9] B. Li, S. R. Tong, C. X. Wang, *et al*. A Method of Design Process Reuse and Optimization Based on Design Process Gene. Advanced Materials Research, 308-310(2011) ,pp.401-404.
[10] J. Chen, S. Y. Zhang, W. H. Bu. Instance feedback of product design based on tentative design chain[J]. Journal of Zhejiang University：engineering Science, 3,44 (2010) ,pp.440-447.
[11] Y. H. Han, K. Lee. A case-based framework for reuse of previous design concepts in conceptual synthesis of mechanisms. Computers in Industry, 4 , 57(2006) ,pp.305–318.

Authors

Jing Chen received the Ph.D. degree in mechanical engineering in 2010 from Zhejiang University, China. She is now a lecturer at school of computer science and technology, Hangzhou Dianzi University. The research interests include manufacturing information and knowledge engineering.

Dewen Seng received the Ph.D. degree from University of Science & Technology, Beijing, China in 2005. He is an Associate Professor and the Vice Director of Software Engineering Institute of Hangzhou Dianzi University. He is now mainly engaged in cloud computing technology and data mining technology.

X. J. Fang received the MS degree in computer software from Hangzhou Dianzi University in 2005. He is now a lecturer at school of computer science and technology, Hangzhou Dianzi University, and is now mainly engaged in machine learning and data mining technology.