INASS

International Journal of
Intelligent Engineering & Systems

http://www.inass.org/

A Product Design Process Modeling Methodology Based on Petri Net

Yongping Hao, Xiaolei Xu, Weiping Shao*, **Chunyan Wang**

R&D Center of CAD/CAM Technology, Shenyang Ligong University, Shenyang 110159, China

* *Corresponding author's Email: shaoweiping3008@sohu.com*

Abstract: In order to realize process modeling and management in the product design, a business process modeling methodology is put forward based on Petri net. In this paper, we merge place node and transition node in Petri net, and reserve token element only by analyzing execution mechanism and dispatching principle of Petri net. In parallel design process instances, a sub-token is introduced, which avoids design process execution conflicts. The definition, dispatching model, and execution mechanism of product design process based on Petri net are discussed respectively. In the prototype implementation section, we developed a Web-based product design process modeling and management system. The system application verifies effectiveness and feasibility of the modeling method.

Keywords: Design process modeling, Petri net, Dispatching model, Execution mechanism, Application prototype

1. Introduction

Today the manufacturing enterprises are faced with a rapidly changing business environment [1]. Varying market demands require frequent change operational and organizational schema. In reaction to these external challenges, the enterprise has to impose internal adaptations in the relevant areas of enterprise operations.

In 1990, Hammer and Champy from MIT and Davenport from Harvard University provided and presented the concept of the business process reengineering [2, 3]. In 1997, Hammer published the Beyond Reengineering, and predicted that the BPM/ BPR would change the enterprise's operation mode and people's working pattern [4].

The Petri net is a kind of combination mode presented by some graphical notation. It is intuitive and understandable, and it has many advantages for describing and analyzing the concurrent features. The Petri net has exact mathematical definition of its execution semantics [5, 6, 7]. Based on mathematics development and its analysis method and technology, the Petri net can be used for static structural analysis and the dynamic behavior analysis [8]. Therefore, the Petri net is an ideal tool for business process modeling. The business process modeling and management mode based on Petri net for product collaborative design process becomes the principal researching field nowadays. Reference [1] describes variation trends of business processes. A structured business process modeling algorithm is presented, and an industry application case is used to illustrate the proposed approach. Petri nets-based workflow analysis techniques can be used to support the verification of modified business processes in polynomial time [9]. Reference [10] proposed an approach to verify workflow processes based on transforming to Petri nets. The approach is a language-independent framework for specific implementation in the context of different modeling languages. Reference [11] proposed a method of composing PWF-nets with workflow patterns based on a timed colored Petri net for workflow management in product design process.

This paper is an improved version of the work presented in [12]. In this paper, via analyzing execution method and dispatching principle of Petri net, a product design process modeling method based on Petri net is established. Merging the place node with the transition node, its dispatching execution mode was improved in product design process modeling. Finally, an application prototype development case is achieved.

2. Question Proposed

A Petri net is usually represented by a 5-tuple:
$\sum = (P,T;F,K,W,M_0)$, where

$PN = (P,T;F)$ is a classical Petri net, it is called basic nets of $\sum$

P is a finite set of Place;

T is a finite set of Transition;

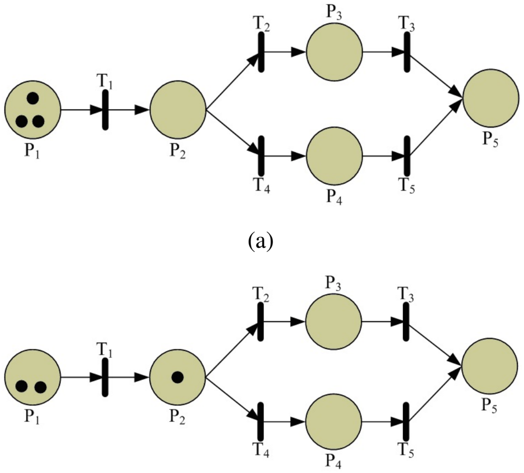
F is the set of Connection, ***F*** can be described as $(P \times T) \cup (T \times P)$;

K is the input function;

W is the output function;

M_0 is the initial mark.

In the Petri net, two kinds of nodes represent the place and transition. The place and the transition are all the non-empty finite set. The Petri net is a bipartite graph, and the places and the transitions alternately run the path formed by arcs. For one transition in the Petri net, if each input place has at least one token (the small black spot in the place is also called token), this transition can be active.



(a)

(b)

Figure 1 Petri net description

In the Figure 1(a), the transition T_1 is ready, when the transition T_1 is enabled, the token will be transited to the place P_2, see Figure 1(b). At this time, the transition T_2 and T_4 are all ready. And so forth, if the T_2 is enabled, the token will be transited to the place P_3. And if the transition T_3 is enabled, the token will be transited to the place P_5 and the whole flow is end.

In the traditional Petri net, the transition is the active element, and by enabling the transition, the process transits from one state to another. However, under some circumstances, this description mode has its own limitation. For the business process management, the mode tends to be huge size, which is not suitable. So, how to simplify the process description is one problem that needs to be resolved in the product design process management.

3. Definition of Design Process Model

In the Petri net, the transition is the active element, usually represents incident, operation, transition or transmission. The place is the passive element, usually representing the medium, buffer, stage or condition. The mark usually presents the dynamic objects in place, which can be the specific matters or abstract information transiting from one place to another place.

One process instance has one token, which is the total token in the flow process. Thus a definition of the business process model is described as:
$\sum = (N,F,K,W,I_0)$,

In which,

N is a node;

F is a set of connections, defined as $N \times N$;

K is the input function;

W is the output function;

I_0 is an initial state.

N is no longer a place node, but the new node that combines the place with the transition nodes. And ***F*** is a set of directed arcs, which represents the execution sequence of each node during the flow process. The model of the business process in Figure 1(a) can be expressed as shown in Figure 2.

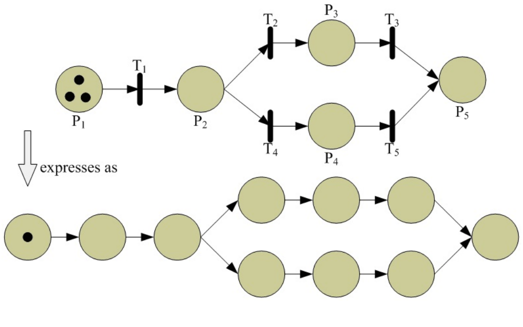


Figure 2 A business process model

It can be seen from the Figure 2, where only one token exists throughout the flow process and no any difference emerges between the place nodes and the transition nodes.

As a result one process flow instance can only define one token. When it comes to the sub node, the problem of sync routing can be resolved by generating the sub mark. For the parallel routing, it needs to establish the sub token (Sub-token), and set up the filiations between token and Sub-token until the end of the sub-flow, then the sub token merges the father-token, and the flow comes back to the main execution route.

Within the design process modeling, the token definition is introduced to present the position of the instance and rapidly locate the process instance. In the process model, the node plays a leading role, representing the medium and buffer of the transition ***T*** and the place ***P***, and it also expresses their instance and operation (including the execution details of the mission). The transformation is only to provide a condition for the transfer, and to play the role connecting two nodes. The dispatch mechanism of the business process is shown in Figure 3.

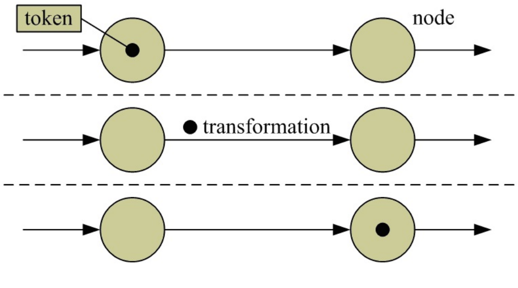


Figure 3 Dispatch mechanism chart

The big circle with the ground color represents the node, and the directed arc with arrows represents transformation (or transition); the small black dot represents the token during flow execution process, its transformation during the process execution process represents the whole process execution route. One process instance has one token, which is the total token in the flow process. Therefore, the definition of the business process flow is:

$\sum = (N;F,K,W,M_0)$, where

N is a node;

F is a set of connection, express as $N \times N$;

K is the input function;

W is the output function;

M_0 is an initial mark.

N is no longer a place node, but the new node that combines the place and the transition nodes. And ***F*** is the set of directed arc, which represents the execution sequence of each node during the flow process. According to these definitions, the product design process flow by Petri net mode is shown in Figure 4.

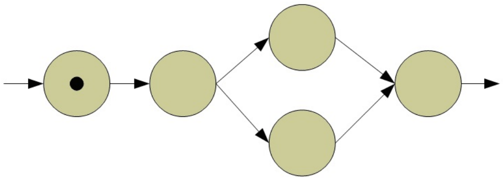


Figure 4 Product design process flow model

As one process instance can only define one token, when a process comes to the sub node, the synchronization routing problem can be resolved by generating the sub mark. For the parallel routing, it needs to establish the sub token (Sub-token), and set up the filiations between token and Sub-token until sub-flow end, then the sub token merges the father- token, the flow comes back to the main route.

If the father-token establishes ***n*** sub tokens, and each sub token executes the corresponding transition, the father-token resides the node of the sub token and remains the state of non-fired. Only when the certain merging condition is met can the father-token be transited to combine node and be re-enabled to finish the mission of the sub and combined node. In the Figure 5 shows a design process mode with sub-process.

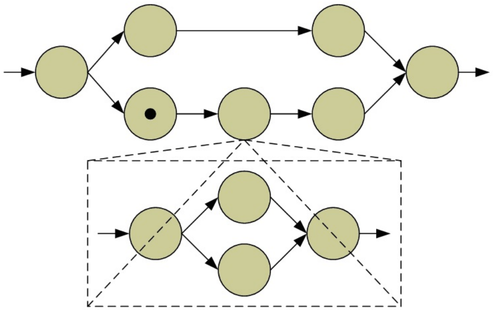


Figure 5 Process model with sub-process

The hierarchical design process model is very complicated in the process management system, when it comes to the complicated process modeling, it is indispensable. By dividing main process into smaller sub-process, we can simplify the complexity of the design process. The sub-process has one advantage that it can permit us to use the pre-defined process. If the sub-process appears many times, it is only to use its definition to simplify the complicated process modeling.

4. Dispatching of Design Process Model

In order to effectively use the node N in the flow process, an authority of decision should be given to the node.

When the user tries to start a flow process, firstly, he needs to build up a process instance, and set up a root mark for this process instance, and put the root mark on the initial node, then go for the next node to put the process instance forward. In Figure 3, when the token receives the transformation order, the instance leaves the node of the present token. During transformation process, the mark puts the token in the corresponding transformation object. But, for the dispatching process, the residing mark has no semantic in the transformation. The object is put into transformation by executing an execution context. And when the Execution Context object is transited from one node to the next, the flow process has been preceded. Figure 6 shows the dispatching model of product design process.

In the actual business dispatching process, the dispatching of the sub and the combined node is not easy, and is used frequently. Here, a sub-flow mode is proposed as shown in Figure 7. The main-flow has only one root-token, and each step is enabled by a certified $token_i$, so that to ensure that the whole flow is under a normal order. Meanwhile, introduce the $token_{ij}$, when the process execution is preceded to $token_i$, it assumes that it encounters the sub-flow, we can use the $token_{ij}$ to enable it and record each execution of each sub-flow. By this method, we can guarantee the execution of each sub-flow, and maintain the filiation's relationship between token and sub-token. When all of the sub-flows are ended, the $token_{ij}$ remerges and continues to execute the $token_{i+1}$. Only one general token is to be used during the whole flow.

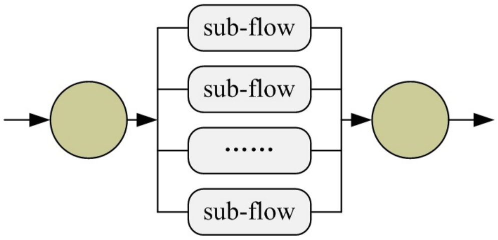


Figure 7 Sub-flow model

5. Execution Mechanism of Design Process

During the business process management mode execution mechanism, we use the token to locate the process instance to the operating node, and the execution of the node is set up according to the transformation of the token. By this method, we can ignore the complicated condition maintenance, and the execution of the node is simpler and understandable. Figure 8 shows the inner condition of the node execution during mode operation.

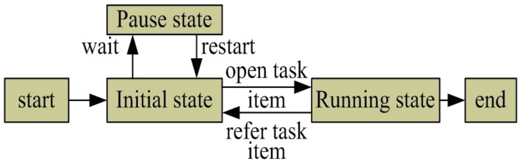


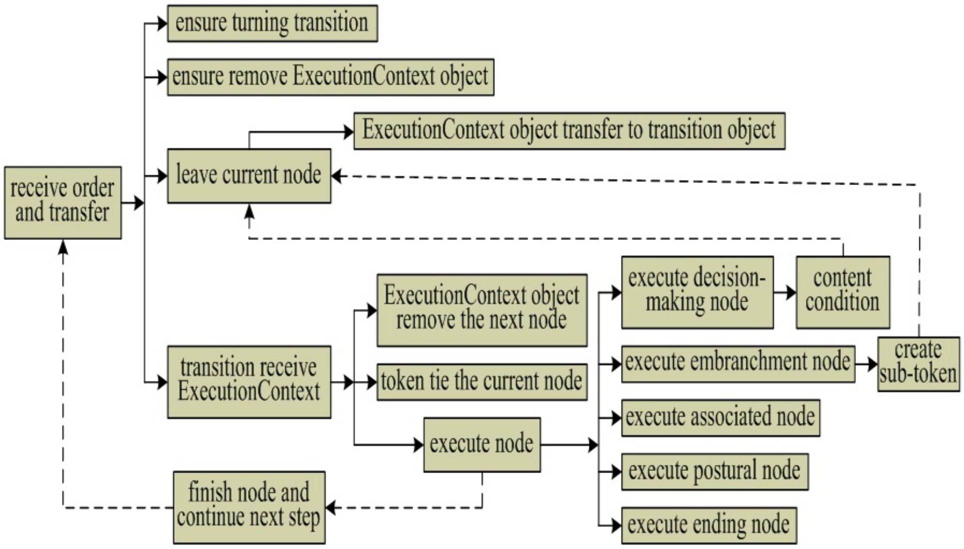
Figure 8 State changing chart during node execution

The serial and parallel execution of the process have defined many types of the nodes, such as auto-executing node, state node, mission node, branch, combination, decision, initial state and the finish node. Their modes of execution are different, but after the process instance, these nodes are at the initial state waiting for additional command from the user. The node mission is at the pause state, once it receives execution mission, it will fire the node. After the mission submission is done, it will send out the token transition order, at this time, the mark will transit to the next node and be combined with the node. Figure 9 shows the mode chart of the node execution and dispatching.

Three methods have been defined for node: they are enter node (enter()), execution node (execute()) and leave node (leave()), and one take() method has been defined in the transformation type. When the token is in the node, the message is sent to the token, the order is to be executed. The message must clarify the transformation of the node for a token (the first transformation is default). When one message is sent to the token, the token obtains its current node, and it calls the Node. leave() method. The node leave() method will generate a node leave incident and call the take() method. The transformation incident is generated from the take () and then call the enter() method on the transformation target node, so as to generate the node enter incident and then call the execute() method. Each type of node has it own action in the execute() method. Therefore, the dispatching and execution of the mode is successfully made.

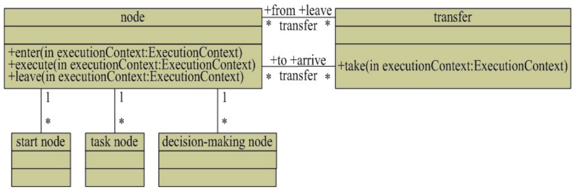
6. Realization of The Business Process Model

According to the drawings design and approval flow, establish the approval flow mode like what is shown in Figure 10. Among which, the $token_1$ is initial;



ensure turning transition
ensure remove ExecutionContext object
ExecutionContext object transfer to transition object
receive order and transfer
leave current node
ExecutionContext object remove the next node
execute decision-making node
content condition
execute embranchment node
create sub-token
transition receive ExecutionContext
token tie the current node
execute associated node
execute node
execute postural node
finish node and continue next step
execute ending node

Figure 6 Dispatching model of design process



node
+from +leave
* transfer *
transfer
+enter(in executionContext:ExecutionContext)
+execute(in executionContext:ExecutionContext)
+leave(in executionContext:ExecutionContext)
+to +arrive
* transfer *
+take(in executionContext:ExecutionContext)
1
*
start node
task node
decision-making node

Figure 9 Mode of process execution and dispatching

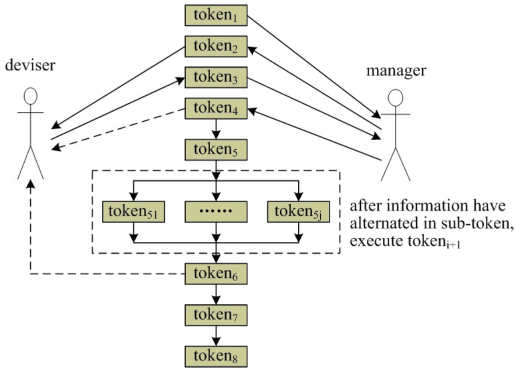


Figure 10 Model of drawings design and approval flow

$token_2$ is design plan; $token_3$ is product design; $token_4$ is design approval; $token_5$ is design mode examination; $token_6$ is drawings countersign; $token_7$ is filing; $token_8$ is end, $token_5 1$ is verification, $token_5 2$ is correction, $token_5 3$ is standard. The flow initiates from $token_1$, and is to be proceeded according to the tokeni increment sequence. In the flow, the symbol (→) represents the steps that must be executed, and the symbol (- -→) represents the steps that can be selected to execute. The sub-flow in the dashed frame has no simple and direct flow relationship with the supervisor and designer, but a complicated and cooperative relationship is connected by a straight line. After all of the sub-flows are finished, the main flow continues to proceed till it finishes.

Use the modeling tools of the business process management system to establish the following mode. In Figure 11, the mode shows the type and name of the node, the personal data and dispatching data that need to be modified are to be configuring red on the attribute panel or the new pop-up window.

Based on the business process mode of the drawings design and approval flow, the feasibility execution is made, and it can be observed during the execution process. The color of the frame of the being executed node is different from other frame. In fact, the execution position of the flow is obtained by inquiring the token of the process instance, so as to change the color of the node symbol. By this method we can check the process execution condition directly.

7. Conclusion

Through the analysis of the execution and dispatching principle of Petri net, we proposed a business process management method based on Petri net. And by redefining and sorting Petri net nodes, the token can reside in the node, so as to define the incident during transformation. Introduce the $token_{ij}$ to the definition of the parallel process token. It does not need to establish the sub token to resolve the sub-flow problem until the parallel routing emerges. Keep the filiations between token and Sub-token until the sub-flow finishes. The sub token combines the father-token and

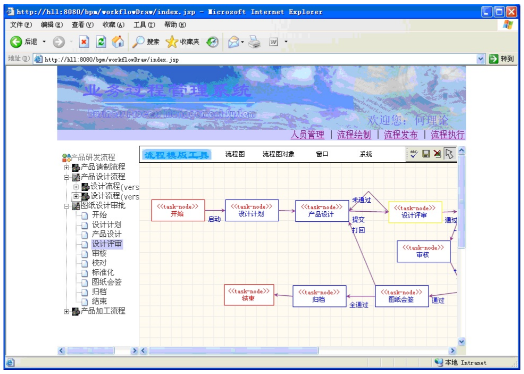


Figure 11 Application realization of design process modeling system

flows back to the main execution route, which ensures that there is only one token in the flow, and there is only one token in the whole flow. There is only one token in the defined process instance because multi-token can conflict during the flow process.

During the business process of the drawings design and approval, check the whole system, which demonstrates the validity of the business process management mode. The definition, dispatching model, and execution mechanism of product design process based on Petri net are discussed respectively. In the prototype system development and implement, we have developed a Web-based process modeling and management system in product design. The system experiences also verifies effectiveness and feasibility of the modeling method based on Petri net.

8. Acknowledgments

This paper is partly supported by the National High-Tech. R&D Program of China (863) (No. 2009AA04-Z167) and the National Natural Science Foundation of China (No.50975183).

References

[1] M. Dong and F. F. Chen, “Petri net-based workflow modeling and analysis of the integrated manufacturing business processes”, *International Journal of Advanced Manufacturing Technology*, Vol.26, pp.1163-1172, 2004.

[2] M. Hammer, “Reengineering work: Don’t AutomateObliterate”, *Harvard Business Review*, Jul. 1990.

[3] K. Jensen, “An introduction to the theoretical aspects of colored Petri net”, In: *A Decade of Concurrency Reflections and Perspectives, Lecture Notes in Computer Science*, Vol.803, pp.230-272, Springer-Verlige, 1994.

[4] Y. Hong, W. Yu and G. Jiao, “Web Service Composition Modeling Based on Colored Petri net”, *International Journal of Systems and Control*, Vol.3, pp.30-34, 2008.

[5] J.X. Li and Z.F. Hou, “Application Petri net in model for Web service composition”, *Application Research of Computers*, Vol.23, pp.149-154, 2006.

[6] Z.J. Wang and Z.X. Cai, “Function/data analysis method based on Petri net”, *Computer Integrated Manufacturing Systems*, Vol.14, pp.1194-1199, 2008.

[7] H.L. Chen, X.Z. Wang and K.C. Zou, “Workflow Process Modeling Based on Petri Net”, *Computer engineering & science*, Vol.30, pp.92-94, 2008.

[8] I.J. Jureta, S. Faulkner, Y. Achbany, and M. Saerens, “Dynamic Web service composition within a service-oriented architecture”, In: *IEEE International Conference in Web Services*, Washington, D.C., USA, pp.304-311, 2007.

[9] WMP van der Aalst, “Verification of workflow nets”, In: *Azema P, Balbo G, (eds) Application and theory of Petri nets, Lecture notes in computer science*, Vol.1248, pp.407-426, Springer, Berlin Heidelberg, 1997.

[10] H.P. Zha, WMP van der Aalst, J.M. Wang, L.J. Wen, and J.G. Sun, “Verifying workflow processes: a transformation-based approach”, *Software and System Modeling*, Vol.10, pp. 253-264, 2011.

[11] S.Ch. Ha and H.-W. Suh, “A timed colored Petri nets modeling for dynamic workflow in product development process”, *Computers in industry*, Vol.59, pp.193-209, 2008

[12] Y.P. Hao, X.L. Xu, W.P. Shao, P.F. Zeng and C.Y. Wang, “Research on Execution Mechanism of Modeling Method for Business Process Based on Petri Net”, In: *2010 Third International Conference on Intelligent Networks and Intelligent Systems*, Shenyang, China, pp.653-656, 2010.