Applied Mathematics & Information Sciences
An International Journal

Dynamic Workflow Modeling Based on Product Structure Tree

Shen Li[1,2]*,XiaoDong Shao*[1,2] *,ZhiHua Zhang*[1,2] *and JianTao Chang*[1,2]

[1] School of Mechano-electronic Engineering, Xidian Univ., Xian 710071, P.R.China
[2] Key Laboratory of Electronic Equipment Structure Design, Ministry of Education, Xidian University. Xian 710071, P.R.China

Received: Jul 6, 2011; Revised Oct. 8, 2011; Accepted Oct. 10, 2011
Published online: 1 September 2012

Abstract: In order to solve the dynamic modification problem of workflow model in product development process,a method of dynamic workflow modeling is presented based on product structure tree (PST).In the method, a dynamic node was introduced into a workflow template, and the refinement rule of a dynamic node was proposed. Through mapping the components of PTS onto the workflow template,the component instances and their affiliation used as refinement input elements,then the dynamic workflow nodes were refined. Validation in the actual engineering case of large-scale antenna development shows that the workflow instance was constructed dynamically at run-time.

Keywords: Product Structure Tree, Dynamic Node, Workflow Refinement Rule, Workflow Instantiating Algorithm.

1. Introduction

Product development process (PDP) is one of the most important business processes for enterprises but it has difficulty in workflow management because of the uncertain and dynamic characteristics. Thus, even though there have been many workflow modeling and management methods, they have limitations to deal with the special characteristics of PDP[1]. For example, utilizing Workflow Management System the large-scale antenna development should integrate hundreds of design links and tasks such as structure design,electrical properties simulation,servo-control, etc.,and achieve real-time transmission of design state and data for the demand of product collaborative design.

Up to the present,most of workflow management systems(WFMSs) are difficult to meet the demand of PDP, and dynamic workflow modeling has been one of the bottleneck problems. At present, there are two main ways for modeling in WFMSs: the first way uses manual modeling, which the workflow activities and their relations are modeled manually. Another way uses automatic modeling with workflow template, which the workflow templates of general business were constructed by administrator of WFMSs and stored in database in advance. When users executed a task, the workflow model was constructed automatically through selecting the workflow template for the task. This way was adopted in the field of office automation(OA) and product data management (PDM), such as render an account in financial system and design approval in product development.

However, the above-mentioned methods cant meet the demands of workflow modeling in PDP compared with OA and PDM, the workflow models for PDP are more complicated. Firstly, the workflow of product development is very complicated. The development of complicate product has hundreds of participants and can produce thousands of design tasks. The relationship between tasks is very complicated. Obviously, the workload of manual modeling is extensive, so it has not engineering feasibility. Secondly, the workflow of product development is a dynamic model. The model is closely related with the product structure which was gradually designed and achieved in PDP. Therefore, the workflow model of product development should be gradual made and refined, it is a dynamic model. So the static workflow template cant meet the dynamic demand.

In fact,much work has been carried out on dynamic workflow modeling [2–9]. The literature[3–5] proposed various methods for flexibility workflow modeling with dy-

* Corresponding author: e-mail: shao_xiao_dong@163.com

namic refinement, but didn't involved the problem product structure influencing on model flexibility, and didn't involved the process knowledge maintenance and reuse. The literature[6–9] proposed automatic generation of workflow processes based on BOM or Product Data Model. However, the workflow model is not available on product development in fact, is a theoretical model.

The above research indicated the method of dynamic workflow modeling, but in practical application, the rules and input elements for model refinement must be formulated. Obviously, there are various rules and input elements in different applying fields. The reports of refinement rules and input elements for workflow modeling in PDP are unknown so far. The aim of this study was to develop a method that would addressed these two problems.

A dynamic workflow modeling method based on product structure tree(PST) is proposed in this paper. In the method, the component instance and their affiliation used as refinement input element, through mapping the PST onto the component workflow template, the dynamic workflow node was refined, then the workflow model for product development was accomplished.

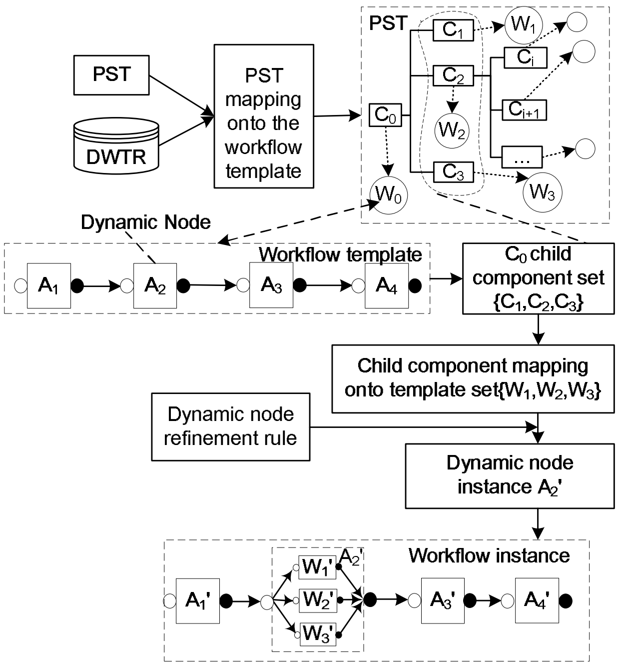

Figure 1 The basic ideas

2. Overview of the approach

The workflow instance is changing with the product structure in PDP, the model cant be constructed with static workflow template in advance. However, the frame of PDP and the workflow of basic component are relatively fixed; the changing factors can be encapsulated with the dynamic node. Then through refinement for the workflow frame containing the dynamic node based on the PST, the workflow instance of the whole product development can be dynamically constructed at run-time.

The method process was shown in Fig.1 The design workflow template repository(DWTR) is a set of workflow model constructed in advance. There are two kinds of workflow template in DWTL, one is general workflow template for basic components, which was stored by the structure and design characteristics; another one is design frame template for the whole products or their complex components, which contained the dynamic node needing be regenerated at run-time. W_0 is a workflow frame template for product $C_0, W_1, ..., W_n$ are general workflow templates, their definition can be seen in section 3.1. The PST is a composite of the part and assembly instance(C_0, C_1, ...,C_n),instance type and their affiliation. In PDP, the DWTL is static, but the PST is dynamically changing.

3. Constructing dynamic workflow template

3.1. Workflow template definition

Workflow representation can take a number of forms, the models include Petri nets, Unified Modeling Language (UML), Business Process Modeling Notation (BPMN), Directed Network Graph (DNG) and their extension forms [10].The workflow templates for special product components were constructed with DNG in this paper. The workflow template is a formal description for design process knowledge and experience, and corresponding to a independent work task. For reusing in workflow modeling, each workflow template has been encapsulated with task node [11] WTN, WTN can be a simple design task or a complicated task comprising several sub-tasks.

A WTN is a tuple $(TName, TType, CType, Pri, Cp, Dp, Act, CtrlF, DataF)$:

$TName$ is an identifying property;

$TType$ is a template classification property;

$CType$ is a component type for mapping onto the component;

Pri is the priority for selecting the template automatically, the value range is 0–9, the bigger the numeric of Pri is, the higher priority the template can be selected;

Cp is a set of control ports, includes CI(input control ports) and CO(output control ports), the value of control port is Boolean, which is deduced by activity data.

Dp is a set of data ports, includes DI(input data ports) and DO(output data ports), each data port represent a data field of design task template, the type of data field can be string , number and data file;

Act is a set of the design activities, $Act = \{A_1, ..., A_j\}$ $(j \geq 0)$, A_j is a sub-node of the WTN;

$CtrlF$ and $DataF$ are the two views(Control Flow and Data Flow[12]) for activity sequence(shown in Fig.2). $CtrlF$ is the object of control flow representing control routing of Act. $CtrF = \{Clink_1, ..., Clink_i\}.Clink$ is a connect line object comprising two control ports of activities, $Clink = \{Cs, Ce, Con\}$, Cs is the start control port of connect line,Ce is the end control port of connect line, Con records the control conditions of connect line, the result of Con is a Boolean value. While Cs and Con are true , the connect line of $Clink$ is enabled. While Con, Cs and Ce are ture, the activity includes Ce is activated. $Clink \subseteq A_{cp} \times A_{cp}$, Acp is the union set of whole Cp;

$DataF$ is the object of data flow, $DataF$=$\{Dlink_1$, ..., $Dlink_i\}$.$Dlink$ is a connect line object comprising two data paorts of activities, $Dlink = \{Ds, De, f\}$, $f : Ds \longrightarrow De$, Ds is the start data port of connect line, De is the end data port of connect line, f is the mapping relation of Ds and De. $Dlink \subseteq A_{dp} \times A_{dp}$, Adp is the union set of whole Dp;

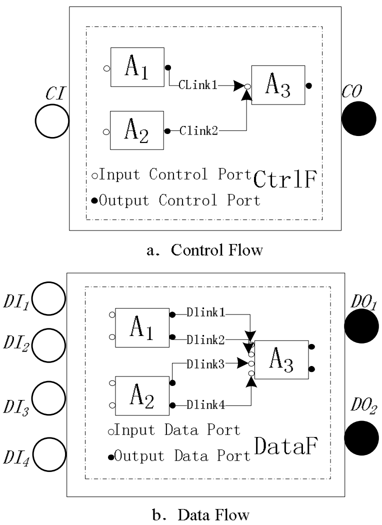

Figure 2 Workflow template node

3.2. Workflow template classification

According to the structure, workflow template nodes can be classified Atom Node(AN), Compound Node(CN) and Dynamic Node(DN), as shown in Table. 1.

(1)AN is the simplest workflow template form, which represents a design task can not be subdivided.

(2)CN is a complex workflow template including several sub-nodes, each sub-node can reference other workflow template node.

(3)DN is a special workflow template form. In PDP, the component of product and assembly is dynamic change, so the workflow template couldnt be constructed in the design time.

Table 1 Workflow template classification.

	$TType$	$CType$	Cp	Dp	Act	$CtrlF$	$DataF$
AN	Atm	$non-null$	$non-null$	$null$	$null$	$null$	$null$
CN	Cpd	$non-null$	$non-null$	$non-null$	$non-null$	$non-null$	$non-null$
DN	Dyn	$non-null$	$non-null$	$null$	$null$	$null$	$null$

4. Mapping workflow templates onto PST

Mapping workflow templates onto PST is the premise to instantiate the frame node. Based on the component type, search the workflow template database and map the matching workflow template onto the node of PST. A PST saves the assembly relationship of product components, so it can be described a tuple (C, R_c). C is a finite set of components, $C = \{C_0, C_1, ..., C_n\}$.$R_c$ is a finite set of components relationship, $R_c = \{< C_0, C_1 >, < C_0, C_2 >, ..., < C_i, C_j >\}, R_c \subseteq C \times C$.

The process of getting the workflow template mapped onto the component C_i was shown in Fig.3,the WTN was extracted from workflow template database filter by the component type $CType_i$. Through cycling treatment, all of mapping WTN for PTS were extracted, then the set (R_{cw}) of component and workflow template relationship was constructed, $R_{cw} = \{< C_1, WTN_1 >, < C_2, WTN_2 >, ..., < C_i, WTN_j >\}$.

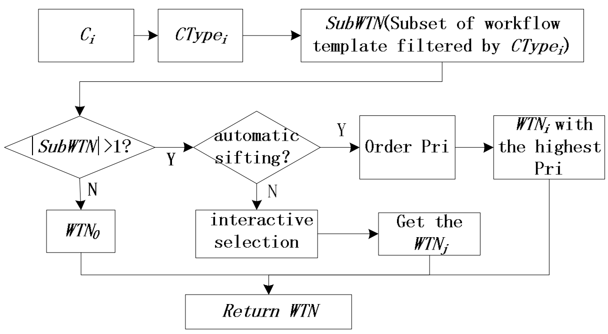

Figure 3 Get the workflow template mapped onto the component

5. Instantiating workflow template

5.1. Workflow instance definition

The workflow instance(WI) is a run-time model constructed for controlling detail PDP based on WTN and instance data. An WI is a tuple $(IName, Chg, PT, Cp, Dp, Act, CtrlF, DataF)$ similar to WTN:

$IName$ is the name of instance, is an identifying feature;

Chg is the user in charge of instance;

PT is the date and time info relating to run instance, includes plan time, current operational progress, the actual completed time, etc.;

$Cp, Dp, Act, CtrlF$ and $DataF$ are the property obtained after instantiating the workflow template.

5.2. Workflow template instantiating algorithm

The instantiation method is various with the different WT-N type. For the general template, the WI can be obtained by directly duplicating the template property. However, for the frame template, it has more complicated process. The algorithm of instantiating WTN_{c0}(the frame template) mapped onto C_0(the root node of PST) was shown in Fig.4.

Step 1.Construct the empty instance (F_{c0}) of WTN_{c0}, judge the sub nodes $TType$ of $WTNco$,If the $TType$ is Atm then goto step(2); else if the $TType$ is Cpd then goto step(3); else if $TType$ is Dyn then goto step(4);

Step 2. Instantiate an AN node: construct the instance F_i , the properties of F_i duplicate the AN node, so the $Cp, Dp, Act, CtrF$ and $DataF$ of F_i are identical with the AN;

Step3. Instantiate an CN node: Construct the empty instance F_i of the CN node, then recursive instantiate the sub nodes of CN node in the control flow order, get the sub instances and insert it into the F_i by order, finally connect the $CtrlF$ and $DataF$ based on the CN node;

Step4. Instantiate an DN node: firstly, Construct the empty instance F_i of the D_N node, then extract sub-set of the children components of C_i . map the WTN_j onto each child component in sub-set and instantiate the WTN_j by order, finally, insert the instances of WTN_j into F_i and configure the $CtrlF$ and $DataF$ of F_i;

Step5. Execute from step(1) to step(4) repeatedly, until each sub node of the WTN_{co} are instantiated. Finally, insert the instances of sub nodes into F_{c0} and configure the $CtrlF$ and $DataF$ based on WTN_{c0}, return F_{c0}.

5.3. Dynamic node refinement

As shown in Table.1, the $Dp, Act, CtrlF$ and $DataF$ in DN are null, the essence of DN refinement is to accomplish the configuration of them in run-time. The steps of DN refinement are as follows: step 1. extract the children components from the current assembly component mapping the DN; step 2. map each child component and obtain the mapped WTN; step 3. instantiate the each mapped WTN and insert the instance into the Act of DN instance; step 4. connect the control and data flow(detail in Section 5.4) ,then obtain the $Dp, CtrlF$ and $DataF$ of DN instance;step 5. substitute the instance for the DN.

The control flow of the WTN mapped onto a certain type antenna structure system design was shown in Fig.5.Among the sub nodes, "Design Component" is a D-N node. To refine the "Design Component", firstly, extract the children components of structure system "Reflector", "Center Part", "Support Beam", "Antenna Pedestal"; secondly, map the $WTN(W_1, W_2, W_3, W_4)$ onto the children components, then instantiate the each mapped WTN and insert the instance of WTN into the instance of "Design Component"; finally, connect the control and data flow of the instance and substitute the instance for "Design Component". While the children components of "Structure System" changed, the DN node in the WTN can regenerate the instance automatically with the PST and ensure the flexibility of WTN in design time.

5.4. Configure instance of dynamic node

The instances of mapped WTN directly inserted into the Act of DN instance had no any control logical and data transmission relations, so the sub instances must be connected reasonably, that is creating the $Dp, CtrlF$ and $DataF$ of the DN instance.

(1) Create Dp

The DI and DO of the DN instance was obtained by union the DI and DO of whole sub instances.

(2) Create $CtrlF$

The $CtrlF$ can be connected by concurrent or serial pattern. In concurrent pattern, the CI ports of all sub instances directly are connected together and the CO ports in the same manner, then the CI and CO value of DN instance are obtained respectively by "And Operation" for the CI and CO value of all sub instances; In serial pattern, the sub instances are connected according to the order of child component mapping relations in PST. The CI of DN instance is same to the first CI of sub instance, and the CO of DN instance is same to the last CO of sub instance.

(3) Create $DataF$

Data connection must base on the control connection[13], so the $DataF$ is relate with the $CtrlF$. In concurrent pattern $DataF$ is null; In serial pattern, generally connect from DI to DO with the same name and type of Dp in the different instances and directly transfer the data value as the default mapping relation, the data mapping relation can be configured in manual.

6. Application

The instantiation steps of the antenna product design workflow template were shown in Fig.6. Area ① shows the PST of a certain type reflector antenna, Area ② shows the

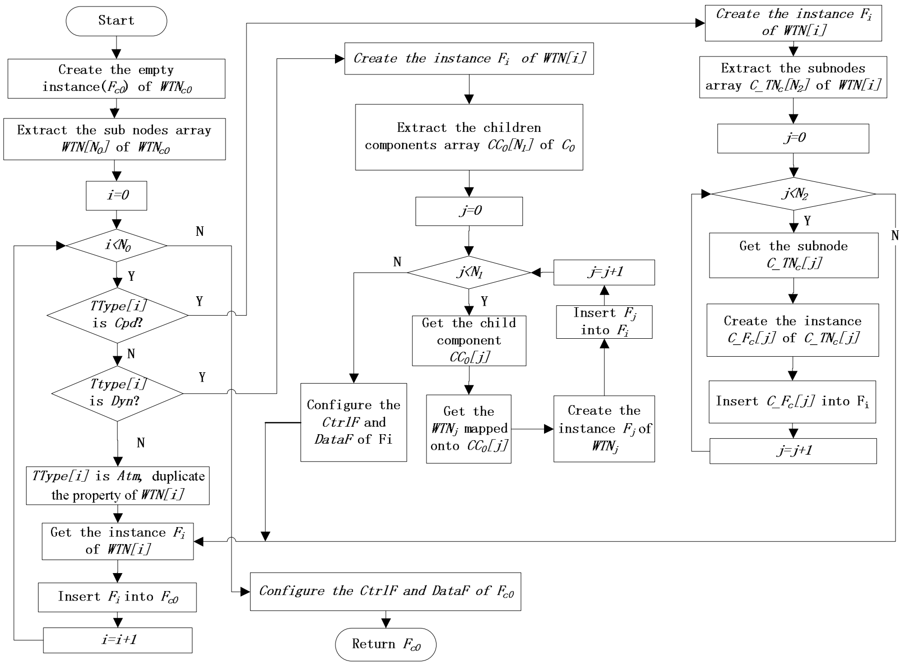

Figure 4 The algorithm flowchart of instantiating a frame template

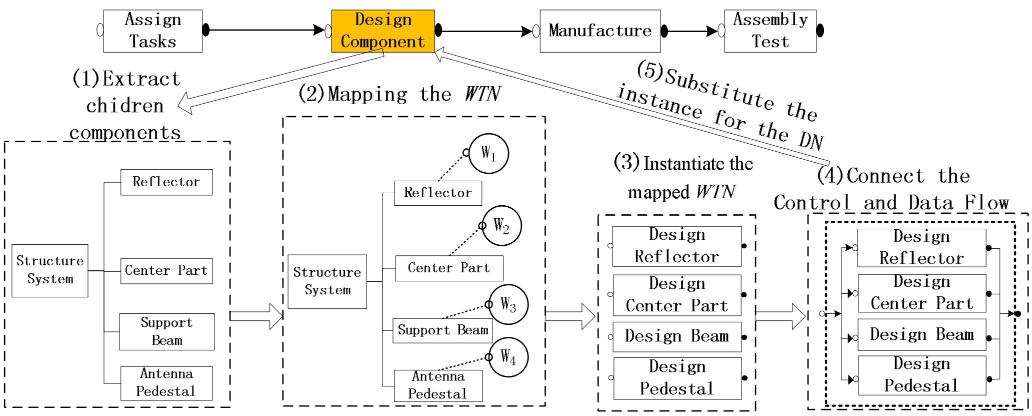

Figure 5 Dynamic node refinement

workflow template WTN_0 control flow of antenna product design; In the WTN_0 , the sub node "Subsystem Design" is a DN node , others are AN nodes ,so the WTN_0 is a frame template. Based on the algorithm, instantiate the each sub node in the WTN_0 by order, after refining the DN node "Subsystem Design", the instance control flow WTN_0 was shown in Area ③ . the parent node of "Subsystem Design" was mapped onto C_0, C_0 has three children components $(C_1, C_2$ and $C_3)$. Through mapping the WTN onto children components, the mapping relations of component and WTN were constructed. As shown in Fig.6, WTN_1 was mapped onto C_1. The instance (F_1) of DN node was refined by sub instances ("Structure System Design"(WTN_1), "RF System Design" and "Servo System Design") connected in parallel. WTN_1 is also a frame template including the DN node "Component Design". Through recursive instantiation of all sub nodes, the instance WTN_0 was constructed. In fact, the instance of antenna product design is large scale, only a small part was shown in Fig.6.

Based on the method in this paper, the Product Design Process Management System (PDPMS) was implemented and the main interface of PDPMS was shown in Fig.7.In PDPMS the WTN database of large-scale antenna struc-

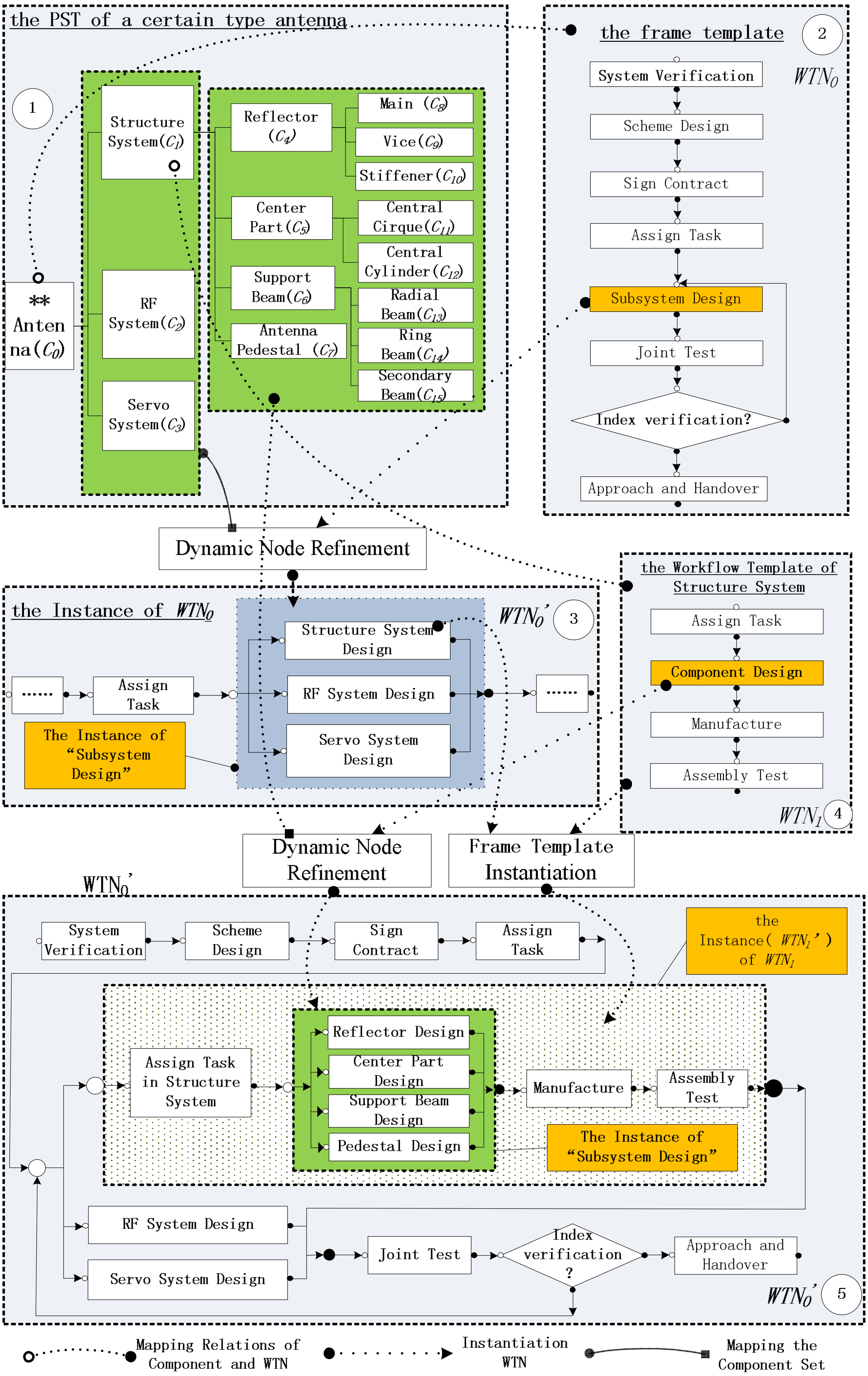

Figure 6 The WTN instantiation of a certain type antenna

ture design was constructed, then the workflow model can be obtained dynamically in run-time based on the PTS.

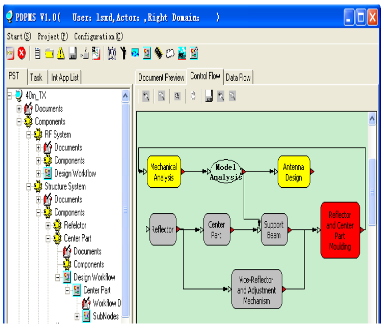

Figure 7 The interface of product design process management

7. Conclusion

In this paper, we have presented an approach to deal with modification of product design workflow with changing of PST. T he workflow template was encapsulated by task node WTN. In run-time, the WTN was mapped onto the node of PST, and the dynamic node was refined based on the PST, finally the instance of workflow was constructed. With the method, the dynamic node in WTN has increased the flexibility of the workflow template and decreased significantly the workload of building the workflow model in manual.

Acknowledgement

The work was supported by the National Natural Science Foundation of China under Grant No. 51035006.the Fundamental Research Funds for the Central Universities, China(Grant No. JY10000904019) and the Fundamental Research Funds for the Central Universities , China(Grant No. 72004181).

References

[1] S. Ha. and H.-W.Suh,Comput.Ind.59.193,209(Elsevier,2008).
[2] J-J.LI,W-P.WANG and F.YANG ,Comput. Integr. Manuf.Sys.8.16.1569,1577(2010).
[3] S. Sadiq, W. Sadiq and M. Orlowska.Pockets of Flexibility in Workflow Specification. Proceedings of the 20th International Conference on Conceptual Modeling:Conceptual Modeling.513,526(Springer-Verlag,Londo,2001)
[4] M.Adams,A.H.M. ter Hofstede,D.Edmond,et al, LNCS.4275.291,308(2006)
[5] L.van Elst, F.-R. Aschoff, A.Bernardi, et al. Proceedings of the 12th International Workshops on Enabling Technologies: Infrastructure for Collaborative Enterprises. 340,345(IEEE,Washington D.C.,2003)
[6] W.M.P. van der Aalst,Comput.Ind.2.39.97,111(Elsevier,1999)
[7] H.A. Reijers, S.Limam and W.M.P. van der Aals. Journal of Management Information Systems.1.20.229,262(2003)
[8] D. MÜller, M. Reichert and J. Herbst. Proceedings of the 15th International Conference on Cooperative Information Systems,131,149(Springer,Berlin,2007)
[9] I.Vanderfeesten, H.A. Reijers and W.M.P. van der Aals. Case Handling Systems as Product Based Workflow Design Support.in Enterprise Information Systems,12.187,198(Springer, Berlin Heidelberg, 2008)
[10] E.Deelman, D. GANNON, M. SHIELDS,et al. Future.Gener.Comp.Sy. 5.25.528,540(2009)
[11] H.Zhuge. Decis. Support Syst. 4.35.517,536(2003)
[12] S.RINDERLE,M.REICHERT,P.DADAM.Distributed and Parallel Databases.16. 91.116(2004)
[13] L.X. ZHAO ,G.F. YIN,B.SHU. Computer Integrated Manufacturing Systems.9.2.112,116 (2003)

Shen Li received the B.Sc. degree in Automation of Industry from the Xidian Unviversity,Xian in 2002, and the M.Sc. degree in Mechano-electronic Engineering in 2005.He is currently a Ph.D.student in the mechanical manufacturing automation. His main research interest on CSCW, modern Integrated Manufacture technology, concurrent engineering, informationize engineering and management, etc.

XiaoDong Shao is a professor,Ph.D.supervisor in School of Mechano-electronic Engineering Xidian University. He is Senior member of China Computer Federation and CAD Secretary General of the association of Shaanxi Province. His main research interests include mechanical CAX ,PDM and CIMS.

ZhiHua Zhang is a mechanical manufacturing automation doctoral student. His main research interests include modern Integrated Manufacture technology, concurrent engineering, informationize engineering and management , etc.

JianTao Chang is a lecturer at the School of Mechano-electronic Engineering, Xidian University. His main research interests include modern Integrated Manufacture technology, decision and support , data mining etc.