2010 3rd International Conference on Computer and Electrical Engineering (ICCEE 2010)
IPCSIT vol. 53 (2012) © (2012) IACSIT Press, Singapore
DOI: 10.7763/IPCSIT.2012.V53.No.2.70

Workflow Model Analysis Based on Colored Petri Nets

Xu Meining+ and Xu Qinglin

School of Information Engineering , Guangdong University of Technology

Guangzhou, China

Abstract—Due to the over-coupling difficulty of workflow modeling and its execution, this paper introduces the importance of independence between resource model and workflow model, constructing unified resource manager, and describes resources using the object-oriented approach. A case based on colored Petri net modeling is given. Combining the Free Choice net, the paper proposes an efficient verifying algorithm which decides if the workflow model is sound. The time complexity of the algorithm is polynomial.

Keywords: workflow model; colored Petri net; resource model; soundness

1. Inodutcion

Workflow model is an abstract representation of the real word business process. The benefits of Workflow model based on Petri net [1] include strict formal semantics, mature analytic technique and graphical nature, so it has been successfully applied to workflow modeling. Colored Petri net (CPN) is formed by the traditional Petri net. CPN has a strong mathematical modeling ability to express the process that makes it possible to describe much more complex business process [2]. We classify resources and construct a unified resource manager in order to improve resource management flexibility. The independence between resource model and workflow model can reduce the coupling of workflow modeling and its execution. In this paper, we deeply analyze soundness of workflow model, and then present an efficient verifying algorithm which decides if the workflow model is sound. The algorithm can be solved in polynomial time.

2. Extended Workflow Net

2.1. Model Mapping

Van der Aalst described twenty workflow patterns in the paper Workflow Patterns [3]. Workflow Management Coalition (WfMC) defined a model which describes the workflow model, called workflow meta model. Figure 1 shows four fundamental structure mapping, including sequential routing, parallel routing, selective routing and iterative routing.

1) *Sequential routing:* We refer to the sequential performance of tasks when these have to be carried out one after another.

2) *Parallel routing:* If more than one task can be carried out at the same time or in any order, then we refer to parallel
routing.

3) *Selective routing:* A process lays down the routing for a specific type of case.

+ Corresponding author.
E-mail address: xumning76@163.com

4) *Iterative routing:* The last form of routing is the repeated execution of a particular task.

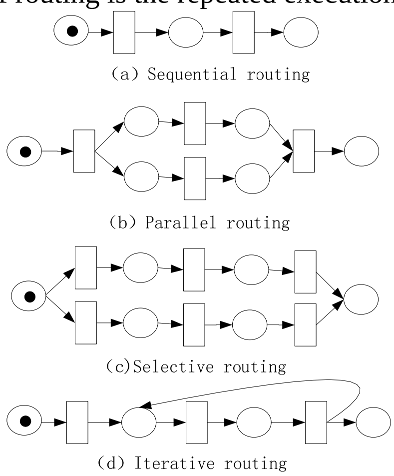

Fig.1. Model mapping

2.2. Introduction of Workflow net and CPN

Aalst addresses the concept workflow net (WF-net) which is based on Petri net. WF-net has two special places, one represents the start of the process, another represents the end of the process. We call them input place (i) and output place (o).

Definition 1: A Petri net PN = (P, T, F) is a WF-net if and only if:

- There is one source place $i \in P$ such that $\cdot i = \Phi$;
- There is one sink place $o \in P$ such that $o \cdot = \Phi$;
- Every node $x \in P \cup T$ is on a path from i to o.

Definition 2: Colored Petri net (CPN) is a 8-tuple CPN = $(\Sigma, P, T, A, C, G, E, I)$, Σ is a finite set of types, called color sets; P is a finite set of places; T is a finite set of transitions; A is a finite set of arcs; C is a color function; G is a guard function; E is an arc expression function; I is an initialization function. The detail definition sees [4].

3. Resource Classification And Model

3.1. Resource Classification

We classify resources in order to improve resource management flexibility [5]. In general, we differentiate between two forms of resource classification: Role and Organization units. A functionally based resource class is known as a role. Resource class can also be classified according to their place in the organization which is called organization units. As depicted in Figure 2, they make the resource class overlapping.

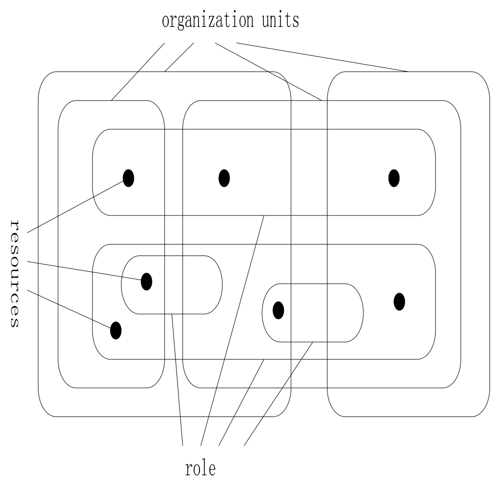

Fig.2. Resource classification

Resource allocation in the workflow is carried out by resource manager according to certain rules. Resource manager is responsible for assignment of resources to workflow. As a result, workflow process model and resource model are independent. We can regard process as client in the C/S structure. Any task in the workflow is executed by a certain resource. When the task is ready to perform, we should request resources from server (resource manager). Resource manager verifies identity at first and searches the related resources in the resource pool if it is right, and then allocates resources. The task can be performed if and only if it has got a related resource. When the task is end, it releases the resource to the resource manager. The resource is added to resource pool and wait for allocation. Resource model is depicted in Figure 3.

Resource model and workflow model should be as independent as possible. That is to say, changes to the resource model (like a new resource is added) should not affect the workflow model. On the other hand, changes to the workflow model (like the addition of an activity) should leave the resource model unaffected, too. At the heart of a workflow system is the workflow engine. This ensures the actual enactment of a specified workflow. One of its core tasks is to allocate work items to resources. In doing so, it must take into account the resource classes specified, as well as such things as separation of function and case management.

The objective of a workflow system is to complete work items as quickly as possible. Some common allocation disciplines are as follows:

5) *First-In, First-Out (FIFO):* If the work items are deal with in the order in which they are created, we refer to a FIFO discipline. FIFO queueing is a simple and efficient allocation rule and is widely used in practice.

6) *Last-In, First-Out (LIFO):* LIFO is the opposite of FIFO. The work items created most recently are deal with first according to this rule.

7) *Shortest Processing Time (SPT):* We can estimate how much time is required to perform an activity, and then select those work items which take the least time. This method can reduce the average time of cases.

8) *Longest Processing Time (LPT):* LPT is opposite of SPT. Sometimes we need to perform those cases.

The other allocation disciplines contain Shortest Rest Processing Time (SRPT) and Earliest Due Date (EDD). There also exist many advanced queueing disciplines.

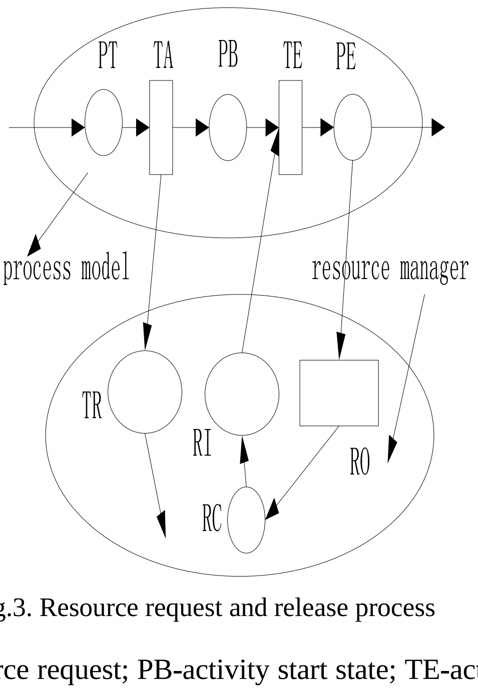

Fig.3. Resource request and release process

PT-activity ready state; TA-resource request; PB-activity start state; TE-activity perform state; PE-activity end state; TR-send request to resource manager; RI-resource; R0-released resource; RC-resource pool.

3.2. Resource Template

We can also describe resources using the object-oriented approach. We should first classify resources according to characteristics of resources, and then define a common resource class which has fundamental nature. Derived classes inherit attributes and behavior of the existing classes. The inheritance relationship of parent classes gives rise to a hierarchy. A resource template is given as follow:

Resource template: Father Resource Class

{
ResID;
ResName;
ResOrgan;
ResAvail;
ResMem;
… …
Description;
}

Father Resource Class is a parent resource class which is inherited by derived class; ResID is used to mark the resource; ResName is the resource name; ResOrgan marks the organization which the resource belongs to; ResCap marks the set of resource ability; ResAvail marks the resource state; ResMem is the resource meta sets; Description is the resource information.

4. Modeling Instance

We can regard drugs production flow as a workflow model. Figure 4 shows a pharmaceutical production and quality monitoring model using Colored Petri net in a pharmaceutical factory. The core modeling process is as follow:

When a pharmaceutical company receives a customer order requirement, we should first check the status flag is right, clearance certificate and instruments is in good condition (activity: production check), and then draw up order intention (activity: order intention) and order contract (activity: order contract). Assessment department works out goods stock estimate (activity: goods stock estimate) based on above two activity and obtains production order. The process enters the materials purchase and test whether they are qualified (activity: materials purchase and check). If the materials have quality problems, we then notify supplier and require returned purchase (activity: not use), else we execute another branch (activity: use available) and begin to produce (activity: production order). Once entering production process, we should track it all the time. If there are some exceptions, stop the production process and enter into exception handling process (activity: exception handing), else enter another branch (activity: monitor ok). Finally, the process is end, qualified products are obtained (activity: finished products). The whole pharmaceutical production process is finished.

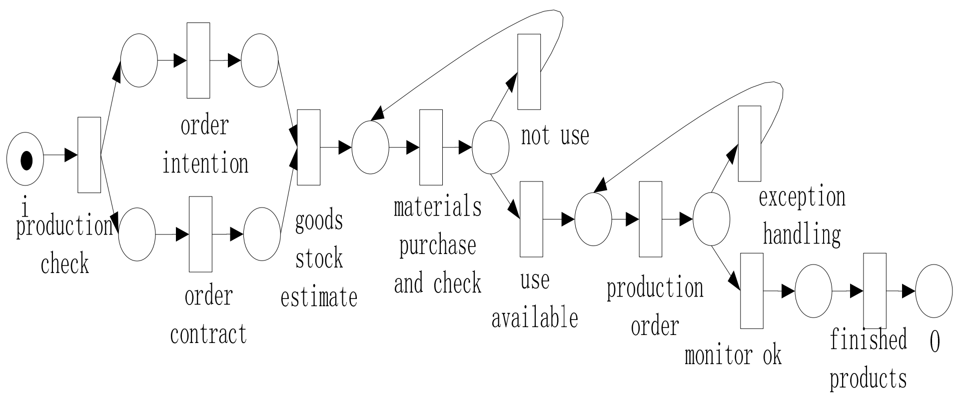

Fig.4. Pharmaceutical production and quality monitoring model

In this model, each transition is corresponding to activity; places are corresponding to enable condition of the activity. During modeling process, Modelers should mark the class to which each transition (activity) belongs and provide the related resources. Each instance (token) of workflow model is colored in running control. The related parameters are above resource template.

5. Structural Soundness Analysis

Static attribute of workflow model based on Petri net can be verified through three requirements stated in definition 1. However a workflow model which satisfies definition 1 still appears some exceptions, such as deadlocks and endless cycles. There is another requirement that should be satisfied:

- For any case, the procedure will terminate eventually and the moment the procedure terminates there is a token in place o and all the other places are empty.
- There should be no dead tasks; it should be possible to execute any task by following the appropriate route through the WF-net.

These two requirements correspond to the so-called soundness property.

Definition 3: A procedure modeled by a WF-net PN = (P, T, F) is sound if and only if:

- For every state M reachable from state i, there exists a firing sequence leading from state M to state o. Formally:

$$\forall_M \left(i \xrightarrow{*} M\right) \Rightarrow \left(M \xrightarrow{*} o\right);$$

- State o is the only state reachable from state i with at least one token in place o. Formally:

$$\forall_M \left(i \xrightarrow{*} M \wedge M \geq o\right) \Rightarrow (M = o);$$

- There are no dead transitions in (PN, i). Formally:

$$\forall_{t \in T} \exists_{M, M'} i \xrightarrow{*} M \xrightarrow{t} M'.$$

WF-net is sound if it satisfies above three requirements. In general, however verifying these three conditions one by one is very complicated. We can check three soundness requirements using a reachability graph. But the construction of the reachability graph for large-scale processes can take up a lot of computer time. What is more, the reachability graph provides little support in repairing a nonsound process definition. Fortunately, a workflow model based on Petri net has strict formal semantics and graphical nature. Soundness corresponds to liveness and boundedness.

Definition 4: A Petri net (PN, M) is live if and only if for every reachable state M' and every transition t, there is a state M'' reachable from M' which enable t.

Definition 5: A Petri net (PN, M) is bounded if and only if $\exists k \in N^+, \forall p \in P, M \in R(N, M_0): M(p) < k$.

Definition 6: The extended Petri net <u>PN</u> = (<u>P</u>, <u>T</u>, <u>F</u>) is defined as follows:

$$\underline{P} = P, \underline{T} = T \cup \{t*\}, \underline{F} = F \cup \{\langle o, t*\rangle, \langle t*, i\rangle\}$$

THEOREM 1: A WF-net PN is sound if and only if (<u>PN</u>, i) is live and bounded [6].

We can check whether <u>PN</u> is live and bounded or not using a reachability graph, but the time complexity is exponential. WF-net based on Free Choice net exists an efficient verifying algorithm that decides if the workflow model is sound. The time complexity is polynomial.

Definition 7: A Petri net is free choice if and only if $\forall t_1, t_2 \in T$, when $\cdot t_1 \cap \cdot t_2 \neq \Phi$, then $\cdot t_1 = \cdot t_2$.

COROLLARY: The following problem can be solved in polynomial time: given a free choice WF-net, decide if it is sound.

PROOF: Let PN be a free choice WF-net. The <u>PN</u> is also free choice. Therefore the problem of deciding whether (<u>PN</u>, i) is live and bounded can be solved in polynomial time [7]. By theorem 1, this corresponds to soundness.

Definition 8: A Petri net PN is well-formed if and only if there is a state M_0 such that (PN, M_0) is live and bounded.

Definition 9: A nonempty subset of places Q in and ordinary net N is called a trap if $Q\bullet \subseteq \bullet Q$, i.e., every transition having an input place in Q has an output place in Q.

Here we present two theorems verifying WF-net is sounded.

THEOREM 2: Rank Theorem [8]

Incidence matrix is N in Free Choice Petri Nets, C_N is the set of all constellations, and <u>PN</u> is well-formed if and only if:

- There exists a nonnegative S_invariant in <u>PN</u>;
- There exists a nonnegative T_invariant in <u>PN</u>;
- Rank (N) = $|C_N|$-1.

THEOREM 3: (as in [9]) Free Choice Net system (<u>PN</u>, M_0) is live and bounded if and only if:

- (<u>PN</u>, M_0) is well-formed;
- All the traps are marked by M_0.

By theorem 2 and theorem 3, there is an efficient algorithm to decide soundness in the Free Choice WF-net. The algorithm is as follow:

Step1. Solve S_invariant in $\underline{PN}$.
Step2. Solve T_invariant in $\underline{PN}$.
Step3. Solve N and C_N in $\underline{PN}$.
Step4. Verify Rank (N) = $|C_N|$-1.
Step5. Verify all the traps are marked by M_0.

If all the steps of the algorithm are satisfied, we can decide WF-net is sound, else it is not sound.

Obviously, this algorithm is not only taken up fewer computers time, but also simple and efficient. Moreover, the time complexity is polynomial. That verifying whether a workflow model is sound or not provides much support in repairing a nonsound process definition. Figure 4 shows a Free Choice Petri net model that meets the requirements, so our model is sound.

6. Conclusion

The paper shows four fundamental structure mapping that WfMC defines in specification. In order to improve resource management flexibility, we classify resources, constructing a unified resource manager and describe resources using the object-oriented approach. Resource model and workflow model should be as independent as possible. Finally, we give a modeling example based on Colored Petri net. We present an efficient verifying algorithm. The algorithm can be solved in polynomial time. The paper only analysises the structure of the workflow model. The follow work is to analysis its performance and capacity planning.

7. Acknowledgment

This work is supported by the 2009 key breakthroughs in key Areas of Guangdong and Hong Kong Project (Foshan) No. 2009Z002.

8. References

[1] W.M.P. Aalst and K.M. Hee, “Workflow Management: Models, Methods, and Systems,” MIT press, Cambridge, MA, 2002.

[2] Kurt Jensen, “Coloured Petri Nets,” Berlin: Springer-Verlag, 1997.

[3] W.M.P. Aalst, A.H.M. ter Hofstede, B. Kiepuszewski, and A.P. Barros, “Workflow Patterns. Distributed and Parallel Databases,” 2003, 14(1): 5-51.

[4] Kurt. Jensen, “Coloured Petri Nets: A High Level Language for System Design and Analysis,” C. Advances in Petri Nets, Berlin: Springer, 1991: 342-416.

[5] Nick.Russell, W.M.P. Aalst, Arthur H.M. ter Hofstede, and David. Edmond, “Workflow Resource Patterns: Indentification, Representation and Tool Support,” vol. 3520/2005 of Lecture Notes in Computer Science, Berlin: Springer, 2005.

[6] W.M.P. Aalst, “Verification of Workflow Nets,” In Application and Theory of Petri Nets 1997, vol. 1248 of Lecture Notes in Computer Science, Edited by P. Azema and G. Balbo, 407-426. Berlin: Springer-Verlag, 1997.

[7] J. Desel, “A Proof of the Rank Theorem for Extended Free-Choice Nets,” Application and Theory of Petri Nets 1992, vol. 616 of Lecture Notes in Computer Science, Edited by K. Jensen. 134-153. Berlin: Springer-Verlag, 1992.

[8] Van Glabbeek R. J and Weijland W. P, “Branching time and abstraction in bisimulation semantics,” J. Journal of the ACM, 1996, 43(3): 555-600.

[9] Jorg Desel and Javier Esparze, “Free Choice Petri Nets,” Vol. 40 of Cambridge Tracts in Thoretical Computer Science.Cambridge University Press, Cambridge, 1995.