Journal of Southeast University (English Edition) Dec. 2002 Vol.18 No.4 ISSN 1003—7985

An Effective Approach to Verify the Correctness of Workflow Process Models Based on Petri Net*

Jiang Hao** Dong Yisheng Luo Junzhou

(Department of Computer Science and Engineering, Southeast University, Nanjing 210096, China)

Abstract: Along with the extensive use of workflow, analysis methods to verify the correctness of the workflow are becoming more and more important. In the paper, we exploit the verification method based on Petri net for workflow process models which deals with the verification of workflow and finds the potential errors in the process design. Additionally, an efficient verification algorithm is given.

Key words: workflow, process modeling, Petri net, verification

Many workflow management systems (WfMSs) are developed based on different paradigms to support workflow management, but few among them provide any form of verification for workflow models. The consequences are that few workflows are thoroughly checked before they are put into practice and often result in errors that have to be corrected at unbearable costs. The paper focuses on the verification of process definition of workflow based on a Petri net approach. We briefly introduce the process modeling of workflow at first. Several correctness issues in workflow process models are discussed. We give a verification method and propose a verification model. Finally we present an effective verification algorithm.

1 Workflow Process Modeling Based on Petri Net

A workflow system contains two basic components: a workflow modeling component and a workflow execution component. The workflow modeling component enables administrators, users and analysts to define, analyze and verify business processes and activities. According to MfMC, the business process is a set of one or more linked procedures or activities which collectively realize a business objective or policy goal. Process model is the representation which consists of a network of activities and their relationships. A workflow process definition (or simply workflow schema) is the formal representation of a business process. In workflow schema, there are several basic constructs such as sequencing, choice, iteration, parallelism and so on. These constructs constitute a meaningful set in which we can write process models.

Petri nets are computer system models for describing asynchronous concurrence and efficient tools to analyze the parallel and concurrence system behaviour. They are an ideal choice of workflow process modeling as they allow easy representation of the logical precondition as the marking and the activities can be represented as transitions. The formal definition of Petri nets is as follows.

Definition 1 (Nets and related definitions) A net is a triple (P, T, F) where:

- P is finite, non-empty set of places, T is finite, non-empty set of transitions;
- $P \cup T = \varnothing$;
- $F \subseteq P \times T \cup T \times P$ is a set of arcs.

Definition 2 (Preset and postset) Let $PN = (P, T, F)$ be a Petri net, for $x \in P \cup T$.

$\cdot x = \{y \in P \cup T \mid \langle y, x \rangle \in F\}$ is the preset of x;

$x \cdot = \{y \in P \cup T \mid \langle x, y \rangle \in F\}$ is the postset of x.

The preset (postset) of a set X is defined as: $\cdot X = \sum_{x \in X} \cdot x \left(X \cdot = \sum_{x \in X} x \cdot \right)$.

Definition 3 (Petri nets and related definitions) A function $M: P \to \mathrm{IN}$ is called marking. A set of places $D \in P$ is marked iff $\exists\, p \in D: M(p) > 0$. A Petri net is a tuple (N, M_0) where N is a net and M_0 is a making named initial marking. $M_0: P \to \mathrm{IN}$.

Let $PN = (P, T, F, M_0)$ be a Petri net, for marking M, if $\forall\, p \in \cdot t: M(p) > 0$, the transition t is said enable in marking M, denoted as $M[t\rangle$.

A transition t enabled at M can fire and thereby

Received 2002-06-17.

* The project supported by the Science Foundation of SEU.

** Born in 1958, male, associate professor.

result in a new marking M' given by

$$M'(p) = \begin{cases} M(p) + 1 & p \in t \cdot \setminus \cdot t \\ M(p) - 1 & p \in \cdot t \setminus t \cdot \\ M(p) & \text{otherwise} \end{cases}$$

this is denoted as $M[t > M'$. The firing sequence $\sigma = t_1, t_2, t_3, \cdots, t_{n-1}$ leading from state M_1 to state M_n is denoted as $M_1[\sigma > M_n$.

A state M_n is called reachable from M_1 iff there is a firing sequence $\sigma = t_1, t_2, t_3, \cdots, t_{n-1}$ such that $M_1[\sigma > M_n$.

A path C from a node n_1 to a node n_k is a sequence $\langle n_1, n_2, \cdots, n_k \rangle$ such that $\langle n_i, n_{i+1} \rangle \in F$ for $1 \leqslant i \leqslant k - 1$. $\alpha(C) = \{n_1, n_2, \cdots, n_k\}$ is the alphabet of C.

A process in workflow is a partially ordered set of activities. Therefore, it is quite easy to map a process onto a Petri net. Activities are modeled by transitions and pre-(or post-) conditions are modeled by places. If there is a token in a place, it represents that the condition denoted as place is ture.

Petri nets which model workflow processes have some interesting characters. Firstly, they always have two special places: begin place p_b and end place p_e which are a source place and sink place, respectively. Secondly, every node in Petri nets should be on a path from p_b to p_e. Finally, every place contains no tokens except the place p_b which has one token at initial marking of the net. These characters can be formalized as follows:

- $|\cdot P_b| = |P_t \cdot| = 0$;
- For $\forall x \in P \cup T$, there exist a path C from p_b to p_e such that $x \in \alpha(C)$;
- The initial marking $M_0 = \{p_b\}$.

The Petri nets with these characters are called workflow nets (WF-net)[1].

2 Correctness in Workflow Process Models

Many up-to-date workflow management commercial products pay more attention to users friendly workflow specification tools, leaving aspects like verification of correctness and error recovery to designers. We believe that the correctness, effectiveness, and efficiency of the business processes supported by the workflow management systems are critical to the business. A workflow process definition which contains error can lead to serious consequences. Flaws in workflow process definition also lead to high throughout times, low service levels, and more need of resources. This is why it is important to analyze a workflow process before it is put into execution. So both designers and users of WfMS see the need for analysis and verification of process definition. Therefore, it is crucial to be able to verify the correctness of the workflow process before it becomes operational.

In our opinions, the correctness of workflow process can be classified into two categories: structural and behavioural. The structural and behavioural correctness are statistic and dynamic respectively. Their detailed implication can be described as follows.

2.1 Structural correctness

- There do not exist any useless activities. A process model has no useless activities if and only if, in WF-net, $\forall t \in T, t \in C$, C is a path from p_b to p_e;
- Any pre-(or post-) conditions in workflow process model are contributed to whole instance execution, i.e., in WF-net, $\forall p \in P, p \in C$, C is a path from p_b to p_e.

2.2 Behavioural correctness

- Deadlock-free There do not exist any deadlocks in process. A deadlock[2] in a process is a reachable state M in which no transition is enable, i.e., $\forall t \in T \Rightarrow \neg M[t\rangle$.
- Properly terminated A workflow process must be properly terminated. There has been a consistency state at the termination of workflow process. That is to say that a workflow process will terminate eventually and at the moment the process terminates that there is only a token in place p_e and all the other places are empty, i.e. $M_e = \{p_e\}$
- Overloaded-free At anytime of workflow execution, every precondition and postcondition must be determined, i.e. there is no overloaded state of process. Overloaded state is the state in which some places contain more than one token. So, no overloaded is equivalent to safety of the nets. In WF-net, overloaded state implies that some conditions of tasks may be tested twice or more at the same time. Safeness is a desirable property, because it makes no sense to have multiple tokens in a place representing a condition. A condition is either ture (1 token) or false (no token).

WF-nets which satisfy conditions mentioned above are called correct. According to Ref.[3], we can conclude that the implication of correctness here is as the same as the soundness property and the simple

control property and it means that WF-net with correctness is live and safe.

Definition 4 (Live, bounded and safe) For giving Petri net (N, M_0):

- The reflexive and transitive closure of reachable markings from marking M_0 is called reachability set and denoted as $R(N, M_0)$
- (N, M_0) is bounded iff $\exists k \in \mathrm{IN}: \forall p \in P, M \in R(N, M_0): M(P) \leqslant k$. (N, M_0) is safe iff it is bounded with $k = 1$. N is structurally bounded iff $\forall M_0 \in [P \rightarrow IN]: \exists k \in \mathrm{IN}: (N, M_0)$ is bounded.
- (N, M_0) is live iff $\forall t \in T, M \in R(N, M_0): \exists M' \in R(N, M)$: t is enabled at M'. N is structurally live iff $\exists M_0 \in [P \rightarrow \mathrm{IN}]: (N, M_0)$ is live.

For the purpose of verification, we define an extended net of WF-net which is obtained by adding a transition t' to WF-net. t' connects p_b and p_e. This extended Petri net is defined as follows.

For giving WF-net $\mathrm{WN} = (P, T, F, M_0)$, it corresponds extended Petri net $\overline{\mathrm{WN}} = (\bar{P}, \bar{T}, \bar{F}, M_0)$, where $\bar{P} = P$, $\bar{T} = T \cup \{t^*\}$ and $\bar{F} = F \cup \{\langle o, t^* \rangle, \langle t^*, i \rangle\})$. The relation between these two nets is illustrated in Fig.1. It is clear that for any marking M' in WN, if the final marking $M_e = \{p_e\}$ is reachable from M', then $\overline{\mathrm{WN}}$ extended from WN is strongly connected.

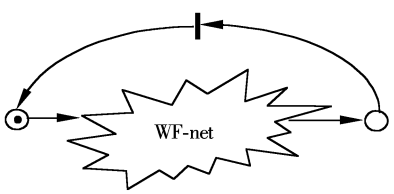

Fig.1 Extended net for WN

From the previous discussion, we can conclude that to guarantee the correctness of workflow processes modeled by WF-net (N, M_0), the sufficient and necessary condition is the liveness and safety of its extended net $\overline{\mathrm{WN}}$.

Conclusion (Correctness of WF-net) A WF-net $\mathrm{WN} = (P, T, F, M_0)$ is correct iff its extended net $\overline{\mathrm{WN}} = (\bar{P}, \bar{T}, \bar{F}, M_0)$ is live and safe.

3 Verification Model

The structural correctness can be checked up by examining the connection of the nets. There are many analysis techniques to verify the behavioural correctness such as reachability tree, reachability graph, coverability graph, T/S-invariant, matrix equation and so on. However, for complex processes, it can be very time-consuming to use these techniques to verify the correctness of WF-net. In fact[4], in ordinary Petri net, the determination problems such as proper termination, safety, liveness, and whether a transition t can be fired, are at least exponential in the worst case by using these techniques. This makes it much more difficult to put the verification procedure into realistic use.

For this reason, we consider a sebset of Petri net — free-choice net. Free-choice net is a class of Petri nets for which strong theoretical results and efficient analysis techniques exist.

Definition 5 (Free-choice net) N is a free-choice net (FC-net) iff $\forall p_1, p_2 \Rightarrow (p_1 \cdot \bigcap p_2 \cdot = \varnothing) \cup (p_1 \cdot = p_2 \cdot)$.

It is easy to see that a process definition composed of AND-splits, AND-joins, OR-splits and OR-joins is free-choice. Clearly, parallelism, sequential routing, conditional routing and iteration can be modeled without violating the free-choice property. If non-free-choice nets are allowed, the choice between conflicting tasks can be influenced by the order in which the proceeding tasks are executed. Because the routing of a case should be independent of the order in which tasks are executed. Non-free-choice is no meaningful in real business processes. In fact, the most meaningful business process is almost free-choice[5]. So, we restrict our workflow nets to free-choice. In the following we only consider free-choice WF-net.

For the sake of building verification model, we give some definitions as follows[6].

Definition 6 Let $N = (P, T, F)$ be a net:

- The incidence matrix $\boldsymbol{C}$ of N is given by $c_{ij} = c_{ij}^{+} - c_{ij}^{-}$, where

$$c_{ij}^{+} = \begin{cases} 1 & (t_i, p_j) \in F \\ 0 & \text{otherwise} \end{cases}$$

and $$c_{ij}^{-} = \begin{cases} 1 & (p_i, t_j) \in F \\ 0 & \text{otherwise} \end{cases}$$

Rank($\boldsymbol{C}$) denotes the rank of the incidence matrix.

- N is an S-graph iff $\forall t \in T$: $|\cdot t| = |t \cdot| = 1$;
- A net $N' = (P', T', F')$ is subnet of N, $(N' \subseteq N)$, iff $P' \in P$, $T' \in T$ and $F' = F \bigcap ((P' \times T') \cup (T' \times P'))$;
- The subset $N' = (P', T', F')$ of N generated by $P' \subseteq P$ is given by $T' = \cdot P' \cup P' \cdot$ and $F' = F \bigcap ((P' \times T') \cup (T' \times P'))$;

- $N' \subseteq N$ is an S-component of N iff N' is a strongly connected S-graph and $T' = {}^{\bullet}P' \cup P'^{\bullet}$;
- N is state machine decomposable (SMD) iff it is covered by S-components, i.e. its every node belongs to an S-component of N;
- $P' \subseteq P$ is a siphon of N iff $P' \neq 0$ and ${}^{\bullet}P \subseteq P^{\bullet}$. A siphon is minimal iff it does not contain a siphon as a proper subset. A siphon P' is strongly connected iff $N' = (P', {}^{\bullet}P', F')$ with $F' = F \cap (((P' \times {}^{\bullet}P') \cup ({}^{\bullet}P' \times P'))$ is strongly connected. Siphons have such characters that if a place $p \in P$ contains no token in marking M', it will still contain no token in any marking reachable from M'.
- The free-choice net (N, M_0) with liveness is safe iff it is covered by strongly connected S-components and there exists only one token in initial marking M_0.

It is clear that the most important property of WF-net checked at the verification of modeled processes is liveness and safety. Liveness corresponds to the absence of global and local deadlocks of modeled system and safety corresponds to absence of overloaded of places. The verification model we proposed is based on the rank theorem.

Theorem 1[7] An FC-net $N = (P, T, F)$ is structurally live and structurally bounded (simply denoted as LBFC) iff N is SMD (i.e. it is covered by S-components) and $\mathrm{Rank}(\boldsymbol{C}) = |P| + |T| - |F \cup (P \times T)| - 1$.

The safety property of WF-net can be simply proved by theorem 2.

Theorem 2[1] A WF-net WN is safe iff its extended net $\overline{\mathrm{WN}}$ is structurally live and structurally bounded.

Therefore, verifying correctness for a given workflow process is converted to check the liveness and safety of its WF-net model and check whether the initial marking of the model is a living marking. According to theorem 1, things we should do in verification procedure are checking strong connectivity for the net, calculating $\mathrm{Rank}(\boldsymbol{C})$, deciding the SMD property of the net and finding a living initial marking.

4 Verification Algorithm

In this section, we discuss the steps of verification algorithm in detail.

1) Checking strong connection of WF-net

The well-known fact is that all the living and bounded Petri nets are strongly connected can be exploited as the precondition. The strong connection of extended net $\overline{\mathrm{WN}}$ of WF-net WN can be checked by traversing the net. If $\overline{\mathrm{WN}}$ is not strongly connected, algorithm terminates with "No". The time complexity of traversing algorithm of the net is $O(|P| + |T| + |F|)$.

2) Checking the condition $\mathrm{Rank}(\boldsymbol{C}) = |P| + |T| - |F \cup (P \times T)| - 1$ is trivial due to well-known algorithm with tolerable time complexity $O(|P|^2|T|)$.

3) The crux of deciding the SMD property of a WF-net is the efficiency of algorithm. Naturally this problem can be solved by searching one S-component containing a place p for any $p \in P$. For simplifying this problem, we use another theorem as follows.

Theorem 3[8] Let $N = (P, T, F)$ be an LSFC-net (LBFC-net). $D \subseteq P$ is a minimal siphon $=> D$ generates an S-component.

This theorem implies that searching for a minimal siphon will result in an S-component. So, the problem to decide the SMD property of a WF-net is converted to find a minimal siphon D containing p and checking D for generating an S-component.

To find a minimal siphon, we can use the algorithm proposed by P. Kemper[7] that has time complexity $O(|P| + |T| + |F|)$.

From definition 6, we can conclude that if $N = (P, T, F)$ is an FC-net with a minimal siphon $D \subseteq P$, D generates an S-component iff ${}^{\bullet}D = D^{\bullet}$ and $\forall t \in {}^{\bullet}D: |t^{\bullet} \cap D| = 1$. That is to say, if a siphon D in net N generates a subnet $\overline{N} = (\overline{P}, \overline{T}, \overline{F})$ which is an S-component, the subnet $\overline{N}$ must be with $\overline{P} = D$, $\overline{T} = {}^{\bullet}D \cup D^{\bullet}$ and $\overline{F} = F \cap (({}^{\bullet}D \times D) \cup (D \times D^{\bullet}))$. To check this condition, we can simply count the arcs $(t, s) \in F$ of any transition $t \in \overline{T}$ leading to an $s \in D$.

Algorithm for checking S component

Input:

T, F of a WF-net $N = (P, T, F), D \in P$ is a minimal siphon of $N, T_D = {}^{\bullet}D \in T$.

Output:

Yes: D generates an S-component; No: D does not generate an S-component.

Initiate:

$\mathrm{num}(x) \leftarrow 0, \forall x \in T - T_D$

$\mathrm{num}(x) \leftarrow 1, \forall x \in T_D$

program:

```
begin
  for each s ∈ D do
      begin
        for each t ∈ T do
          begin
```

```
            if ((t,s) ∈ F)
            then num(t) ← num(t) − 1;
            if (num(t) < 0)
            then return No;
            if ((s,t) ∈ F ∧ t ∈ T − T_D)
            then return No;
        end
    end
  return Yes;
end
```

4) To find a living initial marking, an interesting property of LBFC is given as follows.

Theorem 4[9] Let N be an LBFC-net. M_0 is a living marking iff all minimal siphons are marked at M_0.

Checking existence of unmarked siphon, we can utilize the fact that, in WF-net, it is obvious that the initial marking is

$$M_0(p) = \begin{cases} 0 & p = p_b \\ 1 & \text{otherwise} \end{cases}$$

If D is a siphon in WF-net and $p_b \notin D$, an unmarked siphon is found, algorithm stop with No due to theorem 4, otherwise WF-net is marked living.

We present the outline of an efficient algorithm to decide liveness and safety of workflow nets. It checks whether the net satisfies both properties, liveness and safety, or not. According to theorem 1, we can propose the outline of the verification algorithm as follows.

Input: $(\overline{WN}, M_0)$ where $\overline{WN}$ is the extended net of a WF-net and $M_0 = \{p_b\}$ is the initial marking.

Output: Yes — $\overline{WN}$ is structurally living and structurally bounded and M_0 is a living initial marking, so WN is correct.

No — $\overline{WN}$ is not correct.

Step 1 Checking extended net of WF-net for being strongly connected;

Step 2 Checking Rank($\boldsymbol{C}$) = $|P| + |T| - |F \cup (P \times T)| - 1$;

Step 3 For all places $p \in P$,

① Finding a minimal siphon D containing p;

② Checking D for generating an S-component;

③ Checking if siphon D is unmarked in initial marking M_0.

After successful completion of these steps, all places are covered by at least one S-component and the WF-net is SMD and its liveness at initial marking is ensured.

For determining worst case time complexity of algorithm, worst case time complexity of all steps are: checking strong connection is $O(|P| + |T| + |F|)$; ranking calculation is $O(|P|^2|T|)$; getting minimal siphon is $O(|P| + |T| + |F|)$; checking S-component is $O(|P||T|)$; checking initial marking is $O(|P|)$.

Checking strong connection and calculating ranking are called at most once. The number of calls to get minimal siphon, check S-component and check the existence of unmarked siphon is at most $|P|$. Thus worst case time complexity for verification algorithm is given by

$$\begin{aligned} &O([|P| + |T| + |F|] + |P|^2|T| + |P| \\ &\quad [(|P| + |T| + |F|) + |P||T|] + |P|) \\ &\quad = O(|P|^2|T|) \end{aligned}$$

5 Conclusion

Verification of workflow process models is very important in design time. The major conclusion is that the correctness of workflow process is equivalent to the safety and liveness of WF-net which models the process and the determination of this problem can be solved in polynomial time. The method proposed in this paper can be used in verification of models before putting it into practice, in building models by construction and in correctness preserving at dynamic changing of process models. Therefore, it gives workflow designers a powerful and feasible tool to construct correct process models.

References

[1] van der Aalst W M P. *Structural characterizations of sound workflow nets*[R]. In: Eindhoven University of Technology, 1996.

[2] Onoda S, Ikkai Y, Kobayashi T, et al. Definition of deadlock patterns for business process workflow models[A]. In: *Proc of the Thirty-Second Annual Hawaii Int Conf on System Sciences*[C]. Maui, Hawaii, 1999.

[3] Straub P, Hurtado C A. Control in multi-threaded information systems[J]. *Advances in Computers*, 1996,(45): 1-50.

[4] ter Hofstede H M, Orlowska M E, Rajapakse J. Verification problems in conceptual workflow specifications[J]. *Data & Knowledge Engineering*, 1998, **24**(3):239-256.

[5] Pablo A, Straub, Carlos Hurtado L. Business process behaviour is (almost) free-choice[A]. In: *CESE'96*[C]. Lille, France, 1996.

[6] Yang Wenlun, Yao Shuzheng, Wu Yun. *Software engineering*[M]. Beijing: Publishing House of Electronics Industry, 1997.234-237.(in Chinese)

[7] Kemper Peter. Linear time algorithm to find a minimal deadlock in a strong connected free-choice net[A]. In: *Application*

and Theory of Petri Nets[C]. Berlin: Springer-Verlag, 1992. 319 - 338.

[8] Kovalyov A, Mcleod R D. New rank theorems for Petri nets and their application to Workflow management[A]. In: *Proc IEEE Int Conf on System, Man and Cybernetics*[C]. San Diego, USA, 1998. 226 - 231.

[9] Best E. Some classes of live and safe Petri nets, concurrency and nets[A]. In: *Advances of Petri Nets*[C]. Berlin: Springer-Verlag, 1987. 71 - 94.

一种基于 Petri 网的工作流过程模型正确性的有效验证方法

姜 浩 董逸生 罗军舟

(东南大学计算机科学与工程系, 南京 210096)

摘 要 随着工作流应用的日益广泛,工作流模型正确性验证的重要性日渐突出.论文研究了工作流过程模型的基于 Petri 网的正确性验证方法及发现过程模型设计中存在的潜在错误,给出了一个有效的验证算法.

关键词 工作流, 过程建模, Petri 网, 验证

中图分类号 TP391