Performance Modeling and Analysis of Workflow

JianQiang Li, YuShun Fan, and MengChu Zhou, *Fellow, IEEE*

***Abstract*—Workflow model analysis is performed at logic, temporal, and performance levels. This paper mainly deals with the performance level issues. Workflow net (WF-net) is extended with time information to the timing workflow net (TWF-net). To provide a formal framework for modeling and analyzing workflow, this paper proposes a multidimension workflow net (MWF-net) that include multiple TWF-nets and the organization and resource information. The algorithm to decompose a free-choice and acyclic Petri nets (PN) into a set of T-components is extended to a TWF-net containing iteration structures. Then, resource availability and workload analysis is performed. A method for computing the lower bound of average turnaround time of transaction instances processed in a MWF-net is proposed. Finally a case study is used to show that the proposed method can be effectively utilized in practice.**

***Index Terms*—Performance, Petri nets, turnaround time, workflow model, workload analysis.**

I. Introduction

A BUSINESS process is a set of one or more linked procedures or activities that collectively realize a business objective or policy goal, normally within the context of an organizational structure defining functional roles and relationships [1]. According to Workflow Management Coalition (WfMC) [2], a workflow model is the formal representation of a business process in a form that supports automated manipulation. It includes five relevant perspectives [1]–[3], i.e., process, resource, organization, information, and function perspectives, which are necessary for a workflow that can be automated by workflow management system (WfMS).

Workflow management aims to help business goals to be achieved with high efficiency by means of sequencing work activities and invoking appropriate human and/or information resources associated with these activities. Correctness verification and performance analysis of a workflow model play the key role in implementing successful workflow management.

Workflow model analysis is conducted mainly at three levels, i.e., logical, temporal, and performance levels. Different levels deal with different aspects of a workflow model. Logical level focuses on the correctness verification of event dependency relations in the process control (deadlocks or structural conflicts) [4]–[7], resource allocation [8], and information utilization [9]. Temporal level cares about the interval dependency relations of a workflow model with imposed timing constraints. It mainly includes activity deadline computing [10], [11], temporal inference [12], [13], schedulability analysis and boundedness verification [14], and time violation handling [15], [16]. The logical and temporal levels verification and synthesis can ensure only the functionally working workflow (correctness) but not its operational efficiency. The performance level [17]–[23], [38], on the other hand, focuses on evaluating the ability of the workflow to meet requirements with respect to some key performance indicators such as, maximal parallelism, throughput, service levels, and sensitivity. The analysis of resource availability and utilization, and average turnaround time is performed at this level. Although performance analysis of workflow is an important research topic, it has not captured the attention of many researchers until now [26]. Note that the performance analysis of a workflow model (business process) is different from that of WfMS architecture [24], [25].

The amount of information considered in above three levels increases from logical, temporal, to performance levels. Then, the analysis should be conducted along the same sequence, i.e., from logical, temporal, to performance. At the performance analysis stage, it is assumed that the considered workflow models contain no temporal and logical errors.

According to [26], PN are the only formal techniques able to be used for structural modeling and a wide range of qualitative and quantitative analysis. Formal semantics, local state-based system description, and abundant analysis techniques are three good reasons to use a PN-based workflow management system [27]. Thus, PNs are a naturally selected mathematical foundation for the formal performance analysis of workflow models. Since Zisman [28] used PN to model workflow processes, the PN techniques to study workflow have been widely studied [4], [6]–[10], [14], [20]–[23]. This paper adopts WF-nets [7] as a base mechanism to represent a performance analysis oriented workflow model. According to the semantic properties of workflow models built by most of the workflow modeling tools [7], [29], most enjoy the free-choice characteristic. Thus, this paper deals with only workflow models or WF-nets with free-choice semantics.

Section II introduces some relevant PN subclasses and the concepts of TWF-net by extending the WF-net with time information. Then, the concept of MWF-net is presented. Section III provides an effective algorithm decomposing a free-choice TWF-net into a set of free choice subnets (FC-subnets) by extending a decomposition algorithm [14]. Section IV discusses a method to compute the workload that arrival transaction instances generate for the various resource pools. Section V

Manuscript received February 14, 2003. This work was supported by the National High Technology Research and Development (863) Program of China under Grant 2001AA415020 and the China National Science Foundation under Grant 60274046 and 60228004. This paper was recommended by Associate Editor M. Jeng.

J. Li and Y. Fan are with the Department of Automation, Tsinghua University, Beijing 100084, China (e-mail: lijianqiang@tsinghua.org.cn; fan@cims.tsinghua.edu.cn).

M. Zhou is with the Department of Electrical Computer Engineering, New Jersey Institute of Technology, Newark, NJ 07102-1982 USA and also with the Laboratory of Complex Systems and Intelligence Science, Institute of Automation, Chinese Academy of Sciences, Beijing 100080 China (e-mail: zhou@njit.edu).

Digital Object Identifier 10.1109/TSMCA.2003.819490

presents a method to calculate the lower bound of average turnaround time of transaction instances in a TWF-net. Section VI presents a case study. Finally, Section VII makes conclusions.

II. Basic Concepts

PN is a well-known formalism for describing concurrent discrete event dynamic systems with synchronization [7], [31]. We brief only its basic concepts, to be used in the sequel.

Marked graphs (MG) are PNs such that, $\forall\ p \in P$: $|{}^\bullet p| = |p^\bullet| = 1$, which allow the modeling of concurrency and synchronization, but not conflict. Free Choice nets (FCN) are PN's such that $\forall\ p \in P$: $|p^\bullet| > 1 \Rightarrow {}^\bullet(p^\bullet) = \{p\}$, which allow the modeling of both synchronization and conflict in a restricted and disciplined way, but not of mutual exclusion semaphores [37]. These two subclasses are characterized by their local structural properties. Let $PN = (P, T, F)$ and $PN_1 = (P_1, T_1, F_1)$, PN_1 is a subnet of PN iff $P_1 \subseteq P$, $T_1 \subseteq T$ and $F_1 = F \cap ((P_1 \times T_1) \cup (T_1 \times P_1))$. PN_1 is generated by T_1 iff $P_1 = T_1^\bullet \cap {}^\bullet T_1$ (where the presets and postsets are taken w.r.t. F). It is called a T-component of PN iff PN_1 is the subnet generated by T_1 and, $\forall\ p \in P_1$: $|{}^\bullet p \cap T_1| \leq 1 \wedge |p^\bullet \cap T_1| \leq 1$ PN is called a WF-net iff: 1) PN has two special places: ε and θ where ε is a source place: ${}^\bullet\varepsilon = \phi$ and θ is a sink place: $\theta^\bullet = \phi$; and 2) If we add a new transition t to PN which connects θ with ε, namely, ${}^\bullet t = \{\theta\}$, $t^\bullet = \{\varepsilon\}$, then the resulting extended net $\overline{PN} = (\overline{P}, \overline{T}, \overline{F})$, where $\overline{P} = P$, $\overline{T} = T \cup \{t\}$, and $\overline{F} = F \cup \{(\theta, t), (t, \varepsilon)\}$, is strongly connected. In a WF-net, building blocks such as AND-split, AND-join, OR-split, and OR-join are used to model sequential control structure (SCS), concurrent control structure (CCS), alternative control structure (ACS), and iterative control structure (ICS). Here, ACS has XOR semantics rather than OR semantics [4]. A WF-net has *proper termination* property if starting from the initial state (with only one token in place ε), it is always possible to reach the state with only one token in place θ. A WF-net with proper termination property is *sound* if it has no dead transitions, i.e., for each transition t, it is possible to reach (starting from initial state) a state where t is enabled. As mentioned above, the performance analysis is conducted only after process control logic verification of the model is done, i.e., its soundness is established. Thus, we need to consider only sound WF-nets.

A WF-net models only the process control aspect of workflow. To do the performance analysis, its operational behavior in the enactment environment should be specified. Thus, the extensions of WF-nets with time, resource, and organization information are needed.

Definition 1: A TWF-net is a triple (PN, α, M), where

$PN = (P,T,F)$	WF-net;
$P = \{p_1, \ldots, p_m\}$	set of places representing the state of a transaction instance or the condition of its output transitions;
$T = \{t_1, \ldots, t_n\}$	set of transitions representing activities of workflow;
F	set of directed arcs linking places and transitions, and used to describe precedence relations among activities;
$\alpha = [\alpha(t_1), \ldots, \alpha(t_n)]$	n-dimensional vector where each element $\alpha(t_i)$ denotes t_i's firing delay, i.e., the execution durations of the corresponding activities;
$M: P \rightarrow \{0, 1, \ldots\}$	marking represented in a m-dimensional vector where $M(p_i)$ denotes the number of tokens, i.e., the number of transaction instances in p_i.

Since we only consider free choice TWF-nets, the conflict among transitions can be resolved in a local way, i.e., by specifying the routing rates of tokens at places with several output transitions. Thus, we do not care which firing mechanism (*timed firing* of transitions in three phases or *timed enabling* in single phase firing [32]) is used in the firing process.

Since multiple workflow processes are deployed in an enactment environment and they may share the same set of resource and organization structure, this paper proposes the concept of a multidimension workflow net (MWF-net), which include multiple TWF-nets, organization and resource information, and the necessary perspective mapping relations.

Definition 2: MWF-net is a five tuple (W, O, R, F_P, F_R), where

$W = \{w_1, w_2, w_3, \ldots, w_u\}$	set of TWF-nets;
$O = \{o_1, o_2, \ldots, o_v\}$	set of roles, where o_i is a role defined in the organization perspective;
$R = \{r_1, r_2, \ldots, r_q\}$	set of resource pools, where each r_i denotes a resource pool defined in the resource perspective;
$F_P: T_{w_i} \rightarrow O$	mapping relation between process perspective and organization perspective, where T_{w_i} is a transition set of TWF-net w_i;
$F_R \subseteq O \times R$	binary relation between organization perspective and resource perspective;

We assume in this paper that the firing of a transition (execution of the corresponding activity) needs the support of a specific role. The situation that one transition is projected to several roles can be transformed to this mapping relation by redefining roles and organization structure.

In the framework of MWF-nets, a resource pool is a class of individual resource agents (e.g., software systems or human) that have the same skills and capability and performs the same set of roles. If there are Y_i individual resource agents in each resource pool r_i, we have $Y = [Y_1, Y_2, \ldots, Y_q]$ which is called resource state of the MWF-net in the following discussion. Each role can be allocated to support several transitions' firing and is performed by an individual resource agent of one of the appointed resource pools. Each resource pool may be appointed to undertake many roles. In the enactment environment, the service request (processing of a transaction instance) generated by the firing of a transition is projected (using role as the medium) to one of the individual resource agents that have the probability to be appointed as the performer. Its service time is specified by the firing delay of the corresponding transition.

Fig. 1 shows the graphic representation of an example MWF-net. In the first level, two free choice TWF-nets are used

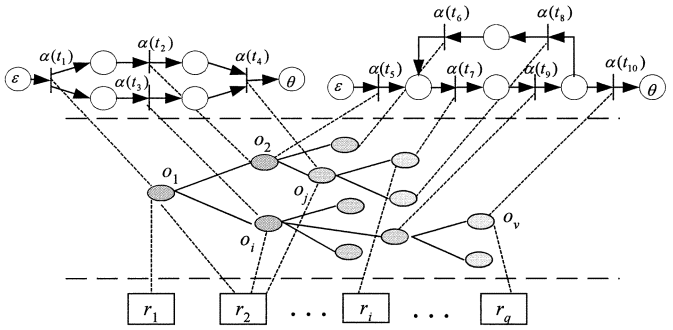

Fig. 1. Graphic representation of an example MWF-net.

to specify the process control and timing perspectives of two workflow models. Firing delay $\alpha(t)$ describes the execution duration of the corresponding activity modeled by t. The second level is the organization perspective, in which each transition is appointed a specific role for its firing (executing the corresponding task). Since the relations among roles are not needed for the performance analysis, they are not specified in MWF-nets. In the resource perspective of the third level, all the resource pools capable of performing the roles of the second level are described. The dotted lines describe the mapping relations among these three perspectives. For example, role o_1 supports t_1's firing (i.e., it generates a request to o_1), the resource agents in resource pools r_1 and r_2 can perform role o_1 (the service requests of o_1 are allocated to r_1 and r_2), and r_2 can perform o_1, o_i, and o_j in the run-time environment.

Each TWF-net in the first level is in fact the control network which describes the routing path of a kind of transaction instances. In the running time environment, the source place ε with token (transaction instance) arrival rate λ can be viewed as a structure of a transition (with no input arcs) with firing rate λ connecting to ε. After the arrival transaction instances are processed or serviced, they depart through sink place θ.

We assume that a transition has exponentially distributed firing delay and transaction instance arrival is a Poisson process. Heuristically, given the transaction instance arrival rates, the routing of transaction instances in the first level can be mapped into the flow of transaction instances among resource pools in the resource perspective of the MWF-net. Then, the flow in the third level of the MWF-net can be modeled as a queuing network, through which the performance related evaluation of the multi-processes workflow may be conducted. We next show how to do performance analysis by using net structure and mapping relations.

III. Model Decomposition of a TWF-Net

Each TWF-net describes the process control and time aspects of a kind of transaction instances. This kind of transaction instances also comprises several types of transaction instances whose routing path in the TWF-net is specified by a subnet of the TWF-net, called FC-subnet for short. This section will demonstrate how to obtain all the FC-subnets by a model decomposition algorithm in which ACSs and ICSs in a TWF-net are investigated.

Since there are no iterations in a workflow model built by the basic process modeling language [5], the ICSs cannot appear in the corresponding TCWF-net [14]. The decomposition algorithm in [14] takes no ICSs into account, and thus cannot apply to a TWF-net with ICSs. In this section, the algorithm in [14] is extended to decompose a free choice TWF-net containing ICSs into a set of FC-subnets, in which there are only ICSs, CCSs and SCSs.

To keep the integrity of this paper, some basic concepts in [14] are explained here. An elementary path in $PN = (P,T,F)$, called a path for short, is $(x_1, x_2, \ldots, x_k)$ such that arc (x_i, x_{i+1}) exists $1 \leq i \leq k-1$, and $x_i = x_j$ implies $i = j$, $1 \leq i, j \leq k$ where $x_i \in P \cup T$. It is called an (elementary) circuit if $x_i = x_j$, $1 \leq i, j \leq k$ implies $i = 1$ and $j = k$. $PN_1 = (P_1, T_1, F_1)$ is a subset of $PN = (P,T,F)$, $path = \{t_0, p_1, t_1, \ldots p_m, t_m\}$ in PN is a transition path of PN_1 iff: (1) It is a path; (2) t_0, $t_m \in T_1$; (3) $p_j \notin P_1$, $1 \leq j \leq m$, and $t_j \notin T_1$, $1 \leq j < m$. Symmetrically, $path = \{p_0, t_1, p_1, \ldots t_m, p_m\}$ in PN is a place path of PN_1 iff: (1) It is a path; (2) $p_0, p_m \in P_1$; (3) $t_j \notin T_1$, $1 \leq j \leq m$, and $p_j \notin P_1$, $1 \leq j < m$. A place p in $PN = (P,T,F)$ is a choice place iff $|p^\bullet| \geq 2$. Suppose $PN_1 = (P_1, T_1, F_1)$ is a path or subnet of $PN = (P,T,F)$ and p' is the first or source place. Given a choice place $p \in P_1$, its choice degree relative to PN_1 is defined as the number of choice places in the path from place p' to p in PN_1. Suppose that R is a path (subnet) set. A choice place p is proper iff $\forall$ path (subnet) in R, p is not on it or p's choice degree is the least.

Next, we extend the algorithm in [14] to decompose a sound free-choice TWF-net into a set R of FC-subnets as follows. We need to use only its structure $PN = (P,T,F)$ in the algorithm.

Algorithm 1 (Decomposition)

Step 1: Construct the extended net $\overline{PN} = (\overline{P}, \overline{T}, \overline{F})$ by adding a new transition t between the source and sink places, i.e., ${}^\bullet t = \{\theta\}$, $t^\bullet = \{\varepsilon\}$.

Step 2: If $P_C = \{p \mid |p^\bullet| \geq 2\} = \phi$, $R = \{\overline{PN}\}$ and go to Step 5 directly. Otherwise, corresponding to an arbitrary choice place $p^1 \in P_C$, a circuit $PN_1 = (P_1, T_1, F_1)$ passing p^1 and t is constructed. In addition, $R_1 = \{PN_1\}$ and $P_S = \phi$, where P_S is a choice place set and R_1 is a subnet set.

Step 3: Repeat the following steps until $\forall\ PN_j = (P_j, T_j, F_j) \in R_1$, $\Psi_j = \{p \mid p \in P_j \wedge |p^\bullet| \geq 2 \wedge p \notin P_S\}$ is empty. Denote the resulting subnet set as $R = \{PN_1, PN_2, \ldots, PN_n\}$

3.1: $R_2 = R_1$, $PL = \phi$;

3.2: For every $PN_j = (P_j, T_j, F_j) \in R_2$, if $\Psi_j = \{p \mid p \in P_j \wedge |p^\bullet| \geq 2 \wedge p \notin P_S\}$ is nonempty, choose the choice place $p \in \Psi_j$ whose choice degree relative to PN_j is the least one in Ψ_j and $PL = PL \cup \{p\}$;

3.3: For every place $p_\alpha \in PL$, if p_α generated from Ψ_k of $PN_k \in R_2$ belongs to

$\Psi_m(m \neq k)$ of $PN_m \in R_2$ and the choice degree of p_α relative to PN_m is not the least one in Ψ_m, then $PL = PL \backslash \{p_\alpha\}$;

3.4: $P_S = P_S \cup PL$, and for every place $p_l \in PL$ do

For every $PN_j = (P_j, T_j, F_j) \in R_2$, if $p_l \in P_j$, there must be a nonempty transition set $\eta = \{t \mid t \in p_l^\bullet \wedge t \notin T_j\}$. Then, $|\eta|$ place paths $path_i = \{p_l, t_i, \ldots, p_{ie}\}$ $(1 \leq i \leq |\eta|)$, where only p_l and p_{ie} belong to PN_j and $t_i \in \eta$, are constructed. If there are c place paths (among the $|\eta|$ place paths) where p_{ie} is located in a path from p_l (not including p_l) to θ, using each of the c place paths to replace the corresponding path from p_l to p_{ie} of the original PN_j respectively, c new subnets $PN_w (1 \leq w \leq c)$ are obtained and added to R_1. For the rest $|\eta| - c$ place paths where p_{ie} is located in a path from ε to p_l (including p_l), merging all of them to PN_j and $PN_w (1 \leq w \leq c)$ in R_1 respectively.

Step 4: Repeat the following exhaustively:

For every $PN_k = \{P_k, T_k, F_k\} \in R$, if there is $t' \in T_k$ with a nonempty set $\{p | p \in t'^\bullet \wedge p \notin P_k\}$, then for every $p_1 \in t'^\bullet P_k$, an arbitrary transition path $path = \{t_0, p_1, t_1, \ldots, p_m, t_m\}$ is constructed and merged into PN_k, where $t' = t_0, t_m \in T_k$. However, if there are choice places in the merged *path* for some new $PN_k s$, all these $PN_k s$ are deleted from R and collected into R'_1, and the following steps are repeated until $\forall\ PN_j = (P_j, T_j, F_j) \in R'_1$, the set $\Psi'_j = \{p' \mid p' \in_j \wedge |p'^\bullet| \geq 2 \wedge p' \notin P_S\}$ is empty. Then the result R'_1 is merged into R.

4.1: $R'_2 = R'_1$, $PL' = \phi$

4.2: For every $PN_j = (P_j, T_j, F_j) \in R'_2$, if $\Psi'_J = \{p' \mid p' \in P_j \wedge |p' \bullet| \geq 2 \wedge p' \notin P_S\}$ is nonempty, each p' must belong to a path from t_0 to t_m. Choose the choice place $p' \in \Psi'_J$ whose choice degree relative to the path from t_0 to t_m is the least one in Ψ'_j, and let $PL' = PL' \cap \{p'\}$;

4.3: For every place $p'_\alpha \in PL'$, if p'_α generated from Ψ'_r of $PN_r \in R'_2$ belongs to $\Psi'_s (r \neq s)$ of $PN_s \in R'_2$ and the choice degree of p'_α is not the least one relative to the path from t_0 to t_m (in Ψ'_s), then $PL' = PL' \backslash \{p'_\alpha\}$;

4.4: $P_S = P_S \cup PL'$, and for every place $p'_l \in PL'$ do

For every $PN_j = (P_j, T_j, F_j) \in R'_2$, if $p'_l \in P_j$, a nonempty post transition set $\eta' = \{t' \mid t' \in p'^\bullet_l (\text{in} PN \wedge t' \notin T_j\}$ exists. Then, $|\eta'|$ place paths $path'_i = \{p'_1, t'_1, \ldots, p'_{ie}\}$ $(1 \leq i \leq |\eta|)$, where only p'_l and p'_{ie} belong to the transition path from t_0 to t_m in PN_j and $t'_i \in \eta$, are constructed. If there are c' paths (among the $|\eta'|$ place paths) where p'_{ie} isn't located in $\{t_0, p_1, t_1 \ldots, p'_l\}$, using each of the c' place paths to replace the corresponding part from p'_l to p'_{ie} of *path* in PN_j, respectively, c' new subnets $PN'_w (1 \leq w' \leq c')$ are obtained and added to R'_l. For the rest $|\eta'| - c'$ place paths where p'_{ie} is located in $\{t_0, p_1, t_1, \ldots, p'_l\}$, merging all of them to PN_j and $PN_{w'} (1 \leq w' \leq c')$ in R'_1 respectively.

Step 5: Deleting transition t and its corresponding arcs (θ, t) and (t, ε) from each subnet in R.

Algorithm 1, which can deal with a workflow model containing ICSs, extends the one in [14]. The extended net is used as the intermediate one for model decomposition. To make Algorithm 1 more readable, Steps 1 and 5 are added to construct and break the extended net by adding additional transition t to PN and deleting t from each extended subnet in resulting R obtained in Step 4. Because PN is sound, the resulting $\overline{PN}$ obtained in Step 1 must be live and bounded. Step 2, corresponding to Step 1 of the old algorithm in [14], is used to construct a circuit PN_1 passing transition t and an arbitrary choice place in PN. Because there are ICSs in PN, we restrict the circuit passing one arbitrary output transition of an arbitrary choice place of PN in Step 1 of the original algorithm in [14] as a circuit that must pass the additional transition t. It guarantees the circuit contains source and sink places, and then it can be processed by the next steps.

Similar to Step 2 of the original algorithm [14], each iteration of Step 3 investigates some choice places of $P_C \backslash P_S$. Steps 3.1–3.3 are equivalent to Steps 2.1–2.3 in the algorithm [14]. Since there are ICSs in PN, however, each $path_i$ of the $|\eta|$ place paths constructed in Step 3.4 may be a part of ICS or ACS, which decides whether a new subnet is generated (if $path_i$ is a part of ACS) or the corresponding subnets are extended (if $path_i$ is a part of ICS). Then, different from Step 2.4 of the one [14], judgment must be made in Step 3.4. Obviously, if p_{ie} belongs to a path from ε to p_l (including p_l), $path_i$ is a part of ICS, otherwise it is a part of an ACS. Since the subnets that need to be extended by the place path of an ICS include PN_j and other c new subnets generated from the c place paths belonging to ACS, Step 3.4 first generates all the new subnets $(c\, PN_w)$ according $c\, path_i$ belonging to ACS and adds them to R_1, and then merges the rest $|\eta| - c\, path_i$ that is a part of ICS to all the corresponding subnets (including the original PN_j and $c\, PN_w$) in R_1. On the one hand, each choice place can only be selected to be a proper choice once, which guarantees that each constructed subnet in Step 3.4 is new for R_1. On the other hand, when a choice place p is put into P_S, the choice places with less choice degree in PN_j must have been in P_S and c subnets $PN_w (1 \leq w \leq c)$ are created for each subnet $PN_j \in R_2$ containing p. Therefore, after Step 3, all the choice places in PN_1 and the place paths constructed in Step 3.4 have been selected into P_S, and all the subnets that can be derived from these choice places are created and added to R. Since there are ICSs in PN, the element in resulting R of

Step 3 may not be a circuit, which is different from Step 2 (all the elements in resulting R are circuits) of the old one [14].

The difference between Step 4 and Step 3 of the old algorithm [14] also lies in the treatment of ICSs in Step 4.4 (corresponding to Step 3.4 in the old one). For each proper choice place selected in each iteration cycle of the Steps 4.1–4.3, there are c' place paths belonging to ACS and $|\eta'| - c'$ place paths belonging to ICS. Correspondingly, c' new subnets are derived first, and then $|\eta'| - c'$ place paths are all merged into PN_j and these c' subnets. After Step 4, $P_S = P_C$ and all the new subnets that can be derived from the choice places in the new merged transition paths in Step 4 are constructed and added into R.

According to the characteristics of a sound TWF-net, it can be proved that the path in Step 4.4 (Step 3.4 is similar to Step 4.4) starting from p'_l must return to another place p'_{ie} in *path*. The reason is that if a *join* place is not in *path*, then the return node of $path'_i$ to PN_j is a place or a transition distinct from t_m. These two situations will destroy the soundness of TWF-net. If the return node is in *path* but not a place, it also conflicts with the soundness of the TWF-net. Hence, the conclusion of the existence of the place path $path'_i$ is derived.

The process control of each kind of transaction instances, which can also be partitioned into a fixed number of types, is specified by a TWF-net. Each element of a TWF-net's decomposition result R corresponds to the routing path of a specific type of transaction instance in the TWF-net. Obviously, the resulting set R of FC-subnets must be determinate. From the analysis mentioned above we know that after a circuit PN_1 passing transition t and an arbitrary choice place is constructed in Step 2, Steps 3–4 can find all the FC-subnets that can be derived from PN. Thus, the arbitrary selection of p^1 and then PN_1 in Step 2 must contribute to the same resulting R.

The complexity of this algorithm is similar to the original one [14] and its discussion is omitted. If the probability of each branch in the ACS is taken into account during the investigation of a choice place, the percentage of the corresponding type of transaction instances accounted for in the total transaction instances processed by the TWF-net can be obtained. In addition, since all the ACSs contained in the TWF-net are disassembled, there are only ICSs, CCSs, and SCSs in each FC-subnet PN_i.

IV. Workload Analysis

This section presents a method to evaluate resource utilization based on a MWF-net. We start with the resource availability analysis of a running time workflow model. Then, we compute resource pools' workload based on FC-subnets and transaction instance arrival rates.

A. Resource Availability Analysis

The failure and repair may happen for a resource agent after it is deployed in the run-time environment. Here failure includes breakdown or downtime for maintenance, and repair includes all recovery steps to get them right and restart the corresponding resource agent. Failures and repairs change the number of available resource agents of a given pool over time.

The technique of continuous time markov chains (CTMC) [33] is employed here to analyze the influence caused by resource agent's transient failure and repair on the resource state of each resource pool, and then on its workload calculation. This basic model implicitly assumes that the time spent in a state is exponentially distributed. Non-exponential failure or repair rates can be accommodated as well, by refining the corresponding state into a set of exponential states [34].

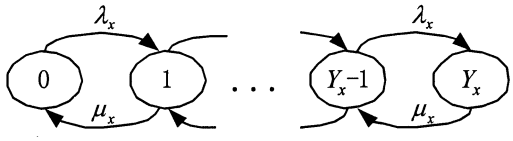

Fig. 2. State transition diagram of resource pool r_x.

Let $Y = [Y_1, Y_2, \ldots, Y_q]$, where $Y_x (1 \le x \le q)$ denotes the number of resource agents in resource pool r_x, and each resource agent has a failure rate μ_x and a repair rate λ_x, which correspond to the reciprocals of the mean time to failure and the mean time to repair, respectively. The CTMC state diagram of resource pool r_x can be obtained as shown in Fig. 2 where state i represents i available resource agents. Obviously, the expected available number of resource agents of r_x in the running time is equivalent to the average queue length of queuing system $M/M/Y_x$[35]. Then, each element of the expected available resource state $Y' = \left[Y'_1, Y'_2, \ldots, Y'_q\right]$ of the MWF-net can be obtained as

$$Y'_x = \begin{cases} \frac{1}{1-\rho_x} - \frac{1+Y_x\rho_x^{Y_x+1}}{1-\rho_x^{Y_x+1}}, & \rho_x = \frac{\lambda_x}{\mu_x} \neq 1 \\ \frac{Y_x}{2}, & \rho_x = 1 \end{cases}$$

B. Workload Computing

Applying the decomposition algorithm of Section III to all TWF-nets in the MWF-net, u FC-subnet sets $R^k (1 \le k \le u)$, each one of which corresponds to TWF-net w_k, can be obtained. Next, we will show how to compute the transaction instance (service request) arrival rate, the service rate, and then the workload of each resource pool.

We first calculate the transaction instance arrival rate of all the transitions in u TWF-nets. The idea is first computing the transaction instance arrival rate of each FC-subnet in every R^k, then using structural analysis to acquire the transaction instance arrival rate of each transition in the FC-subnet.

As mentioned above, each TWF-net in the MWF-net describes the process control and time aspects of a kind of transaction instance which can also be partitioned into several types, and each FC-subnet in R^k of w_k corresponds to a routing path of a specific type transition instance in w_k. Suppose that each w_k includes n_k transitions (then the total number of transitions in the MWF-net is $n = \Sigma n_k$) and is built for the processing of s_k types of transaction instances $\left[I_1^k, I_2^k, \ldots, I_{s_k}^k\right]$. Each type transaction instance I_i^k accounts for a certain proportion of all the transaction instances processed by this workflow process. This kind of information is modeled in the local ACSs of a TWF-net, i.e., by specifying the routing rates of tokens at places with several output transitions. Then we have vector $\beta^k = \left[\beta_1^k, \beta_2^k, \ldots, \beta_{s_k}^k\right]$ satisfying $\beta_1^k + \beta_2^k + \cdots + \beta_{s_k}^k = 1$, where $\beta_i^k \ge 0$ is the percentage of I_i^k in the total number of arrival transaction instances of w_k. Obviously, there should be s_k FC-subnets in R^k, i.e., $R^k = \left\{PN_1^k, PN_2^k, \ldots, PN_{s_k}^k\right\}$.

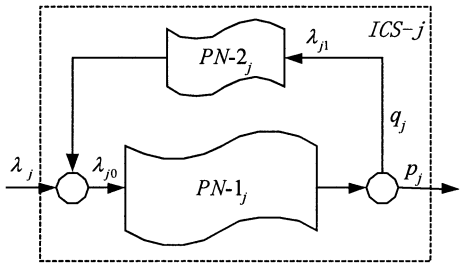

Fig. 3. Example of an ICS that contains $PN-1_j$ and $PN-2_j$.

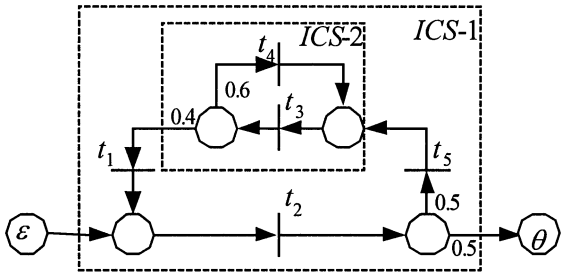

Fig. 4. Example for computing the relative iterative coefficients.

In fact, β_i^k can be obtained by multiplying the appointed probability of all the choice places' output branches belonging to ACS in PN_i^k. Assuming that the average transaction instance arrival rate for w_k is λ^k, the arrival rate of transaction instance of type I_i^k is the ith item of $\lambda_I^k = \lambda^k * \beta$.

An FC-subnet can only have SCSs, CCSs, and ICSs. Obviously, all these control structures can be nested with each other, i.e., a control structure may contain another. But two ICSs or one CCS and one ICS in a sound TWF-net cannot be interlaced with each other. Otherwise the soundness would be destroyed.

PN_i^k corresponds to the routing path of transaction instance I_i^k in the TWF-net. Since there are ICSs in $PN_i^k \in R^k$, a transition belonging to one or more ICSs may fire many times to complete the processing of an arrival transaction instance of type I_i^k. To capture this kind of information, we introduce the relative iterative coefficient that is associated with each transition in PN_i^k. A transition $t \in PN_i^k$ may have a relative iterative coefficient $\Omega_i(t)$ to PN_i^k and a relative iterative coefficient $\delta_j(t)$ to an ICS $ICS\text{-}j \in PN_i^k$. $\Omega_i(t)$ is used to represent t's average firing times to complete the processing of an arrival transaction instance of PN_i^k. $\delta_j(t)$, which is the intermediate to compute $\Omega_i(t)$, is the $\Omega_i(t)$'s constituent multiplicator generated by ICS $ICS\text{-}j$ in PN_i^k.

Now we show how to obtain the relative iterative coefficient of a transition to an ICS $ICS\text{-}j$, and then the iterative coefficient $\Omega_i(t)$ relative to PN_i^k. If t doesn't belong to $ICS\text{-}j$, its relative iterative coefficient to $ICS\text{-}j$ is 1, i.e., $\delta_j(t) = 1$. Otherwise, t's iterative coefficient relative to $ICS\text{-}j$ is determined by its position in $ICS\text{-}j$.

In Fig. 3, $ICS\text{-}j$ of an FC-subnet PN_i^k includes two components $PN\text{-}1_j$ and $PN\text{-}2_j$. To obtain the iterative coefficient of each transaction according to $ICS\text{-}j$, we first compute the transaction instance arrival rates λ_{j0} and λ_{j1} into the two components. Note that in the steady state the departure rates from the two components are also λ_{j0} and λ_{j1}. Arrivals to $PN\text{-}1_j$ are either from the outside at the rate λ_j or from $PN\text{-}2_j$ at the rate λ_{j1}. Therefore, the total arrival rate to $PN\text{-}1_j$ is $\lambda_{j0} = \lambda_{j1} + \lambda_j$. A transaction instance just processed in $PN\text{-}1_j$ goes out of $ICS\text{-}j$ with probability p_j or request $PN\text{-}2_j$ service with probability $q_j = 1 - p_j$. Therefore, the average arrival rate to $PN\text{-}1_j$ is given by $\lambda_{j1} = \lambda_{j0} q$. Thus, $\lambda_{j0} = \lambda_j/(1 - q_j) = \lambda_j/p_j$. and $\lambda_{j1} = q_j \lambda_j / p_j$. Then, the relative iterative coefficients of all the transitions in $PN\text{-}1_j$ to $ICS\text{-}j$ are $1/p_j$, and the relative iterative coefficients of all in $PN\text{-}2_j$ to $ICS\text{-}j$ are q_j/p_j.

Assuming that t belongs to y ICSs of PN_i^k, and its relative iterative coefficient $\delta_x(t)$ to each $ICS\text{-}x$ has been obtained, its relative iterative coefficient to PN_i^k can be calculated as $\Omega_i(t) = \prod_{x=1}^{y} \delta_x(t)$. Obviously, if $t \in PN_i^k$ does not belong to any ICSs of PN_i^k, its iterative coefficient relative to PN_i^k must be 1 (to complete I_i^k's processing, t fires exactly once).

For example, Fig. 4 has two ICSs in the two rectangles of dashed line. Transitions t_1, t_3, t_4, and t_5 belong to the $PN\text{-}2_1$ (from Fig. 3) of $ICS\text{-}1$, their relative iterative coefficients to $ICS\text{-}1$ are same, i.e., $\delta_1(t_i) = 0.5/0.5 = 1 (i = 1, 3, 4, 5)$. Transition t_2 belongs to $PN\text{-}1_1$ (from Fig. 3) of $ICS\text{-}1$, $\delta_1(t_2) = 1/0.5 = 2$. Similarly, $\delta_2(t_3) = 1/0.4 = 2.5$, $\delta_2(t_4) = 0.6/0.4 = 1.5$. Since t_1, t_2, and t_5 do not belong to $ICS\text{-}2$, $\delta_2(t_i) = 1 (i = 1, 2, 5)$. Thus, the relative iterative coefficients of all transitions to this FC-subnet are: $\Omega(t_1) = \Omega(t_5) = 1$, $\Omega(t_2) = 2$, $\Omega(t_3) = \delta_1(t_3)\delta_2(t_3) = 2.5$, $\Omega(t_4) = \delta_1(t_4)\delta_2(t_4) = 1.5$.

To describe distinctly the relative iterative coefficient of a transition to each FC-subnet of w_k, we construct a transition vs. FC-subnet matrix B_k of w_k

$$b_{ij} = \begin{cases} \omega_i(t_j), & \text{if } PN_i^k \text{ includes transition } t_j \\ 0, & \text{otherwise.} \end{cases}$$

Each item b_{ij} of B_k can be seen as the average firing times of t_j to complete the processing of transaction instance I_i^k. Then, the average transaction instance arrival rate of each transition, which belongs to several FC-subnets of w_k, can be obtained by calculating $\lambda_{T_k} = \lambda_I^k * B_k$, where each item of λ_{T_k} represents the average arrival rate of the corresponding transition in w_k.

Next, the information about mapping relations between these two perspectives, i.e., F_P and F_R, will be used to obtain the transaction instance (service request) arrival rate of each resource pool from λ_{T_k} of w_k in the process level.

For each w_k, the firings of n_k transitions generate requests to v roles in the organization perspective. From mapping F_P: $T_{w_k} \rightarrow O$, which specifies distinctly that the firing of a transition in w_k need the support of some roles, a transition vs. role matrix C_k is constructed as follows:

$$c_{ij} = \begin{cases} 1, & \text{if the firing of } t_j \text{ need the support of role } o_i \\ 0, & \text{otherwise.} \end{cases}$$

Since the firing of each transition generates a request to a specific role of the organization, there is only one item being 1 in each column of matrix C_k.

All requests of v roles are allocated to q resource pools $R = \{r_1, r_2, \ldots r_q\}$ in the resource perspective. This kind of information is specified in F_R. However, in the actual configuration of a workflow system, the requests of one role may be allocated to several resource pools. Using γ_{ij} to represent the probability

that the requests of role o_i is allocated to a resource agent of pool r_j, we can construct a role versus resource matrix D as

$$d_{ij}, = \begin{cases} \gamma_{ij} & \text{if } r_j \text{ performs } \gamma_{ij} \text{ part of job appointed to role } o_i \\ 0, & \text{otherwise.} \end{cases}$$

For each resource pool in R, the arrival rate of service requests (transaction instance processing) which are generated from w_k, can be obtained, i.e., the kth item of $\lambda_R^k = \lambda_{T_k} * C_k^T * D$, where C_k^T is the transpose of C_k. The total service request arrival rate is $\lambda_R = \Sigma_k \lambda_R^k$, where the jth item λ_{r_j} represents r_j's transaction instance arrival rate.

The firing delay of t in TWF-nets of the MWF-net is assumed to follow exponential distribution, meaning that each resource agent has exponential service time for the requests from t. As mentioned above, the firings of transitions generate requests from the organization perspective, and they are allocated to different resource pools to be served. However, the requests to one role may be allocated to several resource pools. Then, the service requests of each resource pool may be generated by several transitions, and the transaction instance processing requests coming from one transition may be dispatched to several resource pools. Although the firing delays of all transitions follow exponential distribution, the service time of a resource pool whose requests are generated from many transitions do not in general follow exponential distribution. Yet its average value can be calculated as follows.

First, we construct $C = [C_1 | C_2 | \dots C_u]$, where C_k is a transition vs. role matrix. Then, matrix $A_{n \times q} = C^T * D$ is defined, where n is the total number of transitions in the MWF-net and $a_{ij} = \sum_{l=1}^{q} c_{li} d_{lj}$ the probability that transaction instances (service requests) of $t_i (1 \le i \le n)$ are dispatched (through the perspective mapping) to resource pool r_j. Obviously, t_i may belong to one of u TWF-nets, and the sum of the items in each row is 1. If $\alpha(t)$ represents the firing delay of t, and X_j represents the service time of resource pool r_j, we have $X_j = \alpha(t_1) I_{jt_1} + \alpha(t_2) I_{jt_2} + \cdots + \alpha(t_n) I_{jt_n}$, where I_{jt_i} is the indicator function that takes the value of 1 if resource pool r_j's service request is from t_i; and 0 otherwise. Then the mean firing time of t_i is $E(\alpha(t_i)) = 1/\mu_{t_i}$. Also, $E(I_{jt_i}) = a_{ij} \lambda_{t_i} / \lambda_{r_j}$, where λ_{t_i} is the transaction instance arrival rate of t_i, i.e., the ith term of λ_{T_k}. Then, the average service time of the resource agent in resource pool r_j is

$$\begin{aligned} E(X_j) &= E(\alpha(t_1) I_{jt_1} + \alpha(t_2) I_{jt_2} + \cdots + \alpha(t_n) I_{jt_n}) \\ &= E(\alpha(t_1)) E(I_{jt_1}) + E(\alpha(t_2)) E(I_{jt_2}) \\ &\quad + \cdots + E(\alpha(t_n)) E(I_{jt_n}) \\ &= \sum_{i=1}^{n} \frac{a_{ij} \lambda_{t_i}}{\mu_{t_i} \lambda_{r_j}} = \frac{1}{\lambda_{r_j}} \sum_{i=1}^{n} \frac{a_{ij} \lambda_{t_i}}{\mu_{t_i}}. \end{aligned}$$

Now, the average request arrival rate λ_{r_j} and service rate $\mu_{r_j} = 1/E(X_j)$ of each resource pool r_j are both obtained. Considering the result $Y' = [Y_1', Y_2', \dots Y_q']$ of the resource availability analysis, r_j's workload can be calculated as

$$\rho_{r_j} = \frac{\lambda_{r_j}}{Y_j' \mu_{r_j}} = \frac{\lambda_{r_j} E(X_j)}{Y_j'} = \frac{1}{Y_j'} \sum_{i=1}^{n} \frac{a_{ij} \lambda_{t_i}}{\mu_{t_i}} \text{ and}$$

$$\rho_r = \left[\rho_{r_1}, \rho_{r_2}, \dots \rho_{r_q} \right].$$

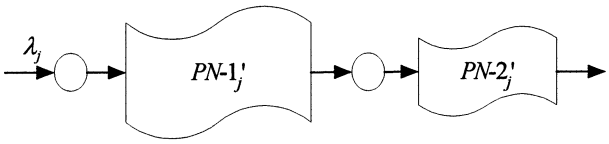

Fig. 5. ICS performance equivalent analysis.

In fact, if $\rho_t = [\lambda_{t_1}/\mu_{t_1}, \lambda_{t_2}/\mu_{t_2}, \dots, \lambda_{t_n}/\mu_{t_n}]$ is constructed for a MWF-net, $\boldsymbol{H} = [h_1, h_2, \dots, h_q] = \rho_t^* A_{n \times q}$ where $h_j = \sum_{i=1}^{n} a_i \lambda_{t_i} / \mu_{t_i}$, is obtained. Then, $\rho_{r_j} = h_j / Y_j'$ and computing each resource pooal's transaction instance arrival rate can be avoided. Since only correct and effective MWF-nets are considered, a MWF-net must be bounded, then, each element in ρ_r is smaller than 1. Hence, the resource pool with the maximal value of ρ_{r_j} is the bottleneck of workflow.

V. Lower Bound of Turnaround Time

The turnaround time, one of the most important performance indicators in industry, of a specific transaction instance is the total amount of time spent from the start to end of handling a transaction instance. Clearly, the turnaround times of different types transaction instances sharing the same TWF-net may be different. We next derive the lower bound of the average turnaround times of transaction instances, which reflects the shortest response time of a workflow system from the viewpoint of expectation and can provide valuable information for system designers.

A transition's delay corresponds to the service time of a service station (resource pool) in a queuing system. However, the sojourn time of a transaction instance in a service station (resource pool) consists of both delay of the transition and waiting time in the corresponding queue. It is larger than the service time in general. The turnaround time of the transaction instance in a workflow is the sum of all sojourn times in all the visited resource pools along its routing path. Under the assumption of infinite servers, if a transaction instance is given the highest priority (e.g., in an emergency treatment), it is served immediately when it arrives at a service station, i.e., its waiting time is negligible. Based on this idea, the performance equivalent structure analysis is conducted to derive the lower bound of a transaction instance's average turnaround time.

Since the decomposition algorithm has applied to all u TWF-nets in a MWF-net, a union $\Re$ of u FC-subnet sets $R^k (1 \le k \le u)$ can be obtained. Each element of $\Re$ corresponds to a processing path of a transition instance type. Next, corresponding to ICSs in a FC-subnet, a performance equivalence theorem is presented.

Suppose that the ICS shown in Fig. 3 is an innermost ICS, i.e., it does not contain other ICS inside. The performance equivalent SCS of this ICS is given in Fig. 5. $PN\text{-}1_j'$ and $PN\text{-}2_j'$, which have the same net structure as $PN\text{-}1_j$ and $PN\text{-}2_j$ respectively, are two constituent components of the SCS.

Theorem 1: If the firing delay of $t_i(t_j)$ in $PN\text{-}1_j$ $(PN\text{-}2_j)$ of Fig. 3 has exponentially distributed random delay with a mean rate $\mu_{t_i}(\mu_{t_j})$, the corresponding $t_i'(t_j')$ in $PN\text{-}1_j'(PN\text{-}2_j')$ of Fig. 5 has a mean firing rate $p_j \mu_{t_i} (p_j \mu_{t_j} / q_j)$.

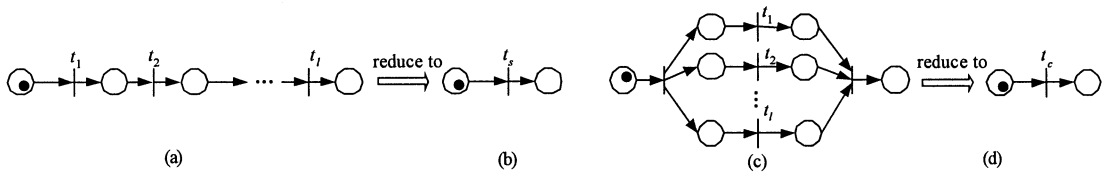

Fig. 6. SCS and CCS performance equivalent analysis.

Proof: There are only SCSs and CCSs in an innermost ICS, and then the two constituent components $PN\text{-}1_j$ and $PN\text{-}2_j$ in the ICS of Fig. 3 must be structurally acyclic MGs. To complete the processing of each transaction instance, the number of firing times of all the transitions in $PN\text{-}1_j(PN\text{-}2_j)$ is the same. A transaction instance processed in $PN\text{-}1_j$ next requests $PN\text{-}2_j$ with probability q_j. If a transition t_i in $PN\text{-}1_j$ fires n times with probability pq^{n-1}, a transition in $PN\text{-}2_j$ must fire $n-1$ times, and the total firing delay of this iterative structure is equivalent to that of a sequential structure consisting of n $PN\text{-}1_j$ and $n-1$ $PN\text{-}2_j$. Since the mean firing delay of $t_i(t_j)$ in $PN\text{-}1_j(PN\text{-}2_j)$ is $1/\mu_{t_i}(1/\mu_{t_j})$, the mean firing delays of the equivalent transition t'_i in $PN\text{-}1'_j$ and t'_j in $PN\text{-}2'_j$ can be calculated as

$$\begin{aligned}\frac{1}{\mu'_{t_i}} &= \sum_{k=1}^{\infty} p_j q_j^{k-1} \frac{k}{\mu_{t_i}} = p_j \sum_{k=1}^{\infty} q_j^{k-1} \frac{k}{\mu_{t_i}} \\ &= p_j \frac{1}{p_j^2} \frac{1}{\mu_{t_i}} = \frac{1}{p_j \mu_{t_i}} \\ \frac{1}{\mu'_{t_i}} &= \sum_{k=1}^{\infty} p_j q_j^{k-1} \frac{k-1}{\mu_{t_i}} = p_j \sum_{k=1}^{\infty} q_j^{k-1} \frac{k-1}{\mu_{t_j}} \\ &= p_j \frac{q_j}{p_j^2} \frac{1}{\mu_{t_j}} = \frac{q_j}{p_j \mu_{t_j}}\end{aligned}$$

Therefore, Theorem 1 holds. □

Note that the equivalence between the two control structures does not hold with respect to the distribution function of the firing delay [33]. The firing delay of the transition in the equivalent SCS is in fact the product of original transition's relative iterative coefficient to the ICS and its firing delay. Since the firing delay of all the transitions in the original ICS follows exponential distribution, the firing delay of the transition in the equivalent SCS follows either gamma if the relative iterative coefficient takes nonintegral value or Erlang distribution if the relative iterative coefficient takes integral value.

Theorem 2 [20]: Assume that $t_i(1 \le i \le l)$ has exponentially distributed delay with mean $1/\mu_{t_i}$ and others have zero delay in Fig. 6(a) and (c). A SCS (CCS) performance equivalent transition $t_s(t_c)$ in Fig. 6(b) and (d) has mean value

$$\begin{aligned}\frac{1}{\mu_{t_s}} = \sum_{i=1}^{l} \frac{1}{\mu_{t_i}} \Bigg(\frac{1}{\mu_{t_c}} &= \sum_{i=1}^{l} \frac{1}{\mu_{t_i}} - \sum_{i<j} \frac{1}{\mu_{t_i} + \mu_{t_j}} \\ &+ \sum_{i<j<k} \frac{1}{\mu_{t_i} + \mu_{t_j} + \mu_{t_k}} \cdots + (-1)^{l-1} \\ &\times \sum_{i1<i2<\cdots<il} \frac{1}{\mu_{t_{i1}} + \mu_{t_{i2}} + \cdots + \mu_{t_{i1}}} \Bigg).\end{aligned}$$

The firing delays of t_s and t_c follow hypo and hyper- exponentially distributions respectively. According to [33], both can be approximated as an exponentially distributed random variable. Since both Erlang and gamma distributions are special cases of hypo-exponential distribution, we can also approximate the firing delay of a SCS-equivalent transition in Theorem 1 as an exponentially distributed random delay. Then, using these theorems repetitively under the principle of *innermost control structure first*, we can reduce a FC-subnet in $\Re$ into a subnet with only places ε and θ and one performance-equivalent transition. Each type transaction instance corresponds to such a reduced subnet whose transition's mean delay represents the lower bound of the corresponding type transaction instance's average turnaround time in its corresponding TWF-net.

VI. A Case Study

Suppose that a MWF-net contains two TWF-nets w_1 and w_2 as shown in Fig. 7, in which "[x]" attached to each transition of a TWF-net denotes the mean value of its exponential firing delay. The workflow has 5 roles $O = \{o_1, o_2, o_3, o_4, o_5\}$ and 6 resource pools $R = \{r_1, r_2, r_3, r_4, r_5, r_6\}$. The mapping relations are described in F_P and F_R. The resource state of the MWF-net is $Y = [3, 2, 8, 4, 2, 2]$. Given the failure rate of all the resource agents $\mu = 0.1$, and repair rate $\lambda = 0.95$, by using resource availability analysis method, we obtain the available resource state $Y' = [2.8828, 1.8859, 7.8824, 3.8826, 1.8859, 1.8859]$.

To realize the workload analysis, we should decompose each TWF-net in the MWF-net into a set of FC-subnets. After Step 1 of Algorithm 1 is applied to the two TWF-nets, their corresponding extended nets are obtained in Fig. 7(c) and (d).

Fig. 8 illustrates the sequential model decomposition process of the extended w_1 in Fig. 7(c). After Step 2, in which $p^1 = p_A$, the circuit PN_1 in Fig. 8(a) is obtained. For Step 3, initially, $P_C = \{p_A\}$, $P_S = \phi$, and $R_2 = R_1 = \{PN_1\}$. After Step 3.3, $PL = \{p_A\}$. Obviously, p_A is a proper choice place and merged to P_S . $\eta = \{t_{A3}\}$ in Step 3.4. The only place path $path_1 = \{p_A, t_{A3}, t_{A6}, p_B\}$ where p_A and p_B belong to PN_1, is constructed. Since p_B belongs to a path from p_A to θ, meaning the place path being a part of an ACS, a new subnet PN_2 shown in Fig. 8(b) is obtained by replacing the original path from p_A to p_B in PN_1 with $path_1$ and added to R_1. Step 3 stops after one iteration. Now, $P_s = \{p_A\}$ and $R = R_1 = \{PN_1, PN_2\}$. Next, apply Step 4 to the resulting R of Step 3. Since t_{A_2}'s output place doesn't belong to PN_1, for which transition path $\{t_{A2}, t_{A5}, p_c, t_{A9}, t_{A10}\}$ is constructed and merged into PN_1, and then PN_1 in R is replaced with the subent in Fig. 8(c). Similarly, transition path $\{t_{A3}, t_{A7}, p_c, t_{A9}, t_{A10}\}$ is constructed and merged into PN_2, then, the subnet in Fig. 8(d), which replaces PN_2 in R, is obtained. Because there is no new choice place

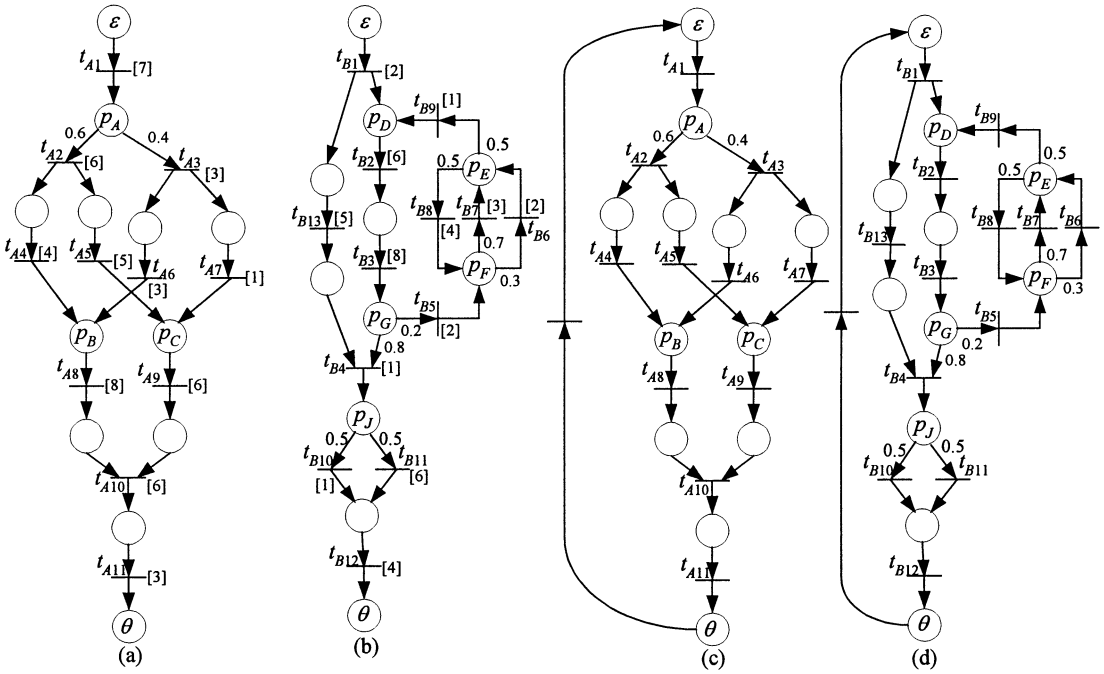

Fig. 7. TWF-nets and their extended nets.

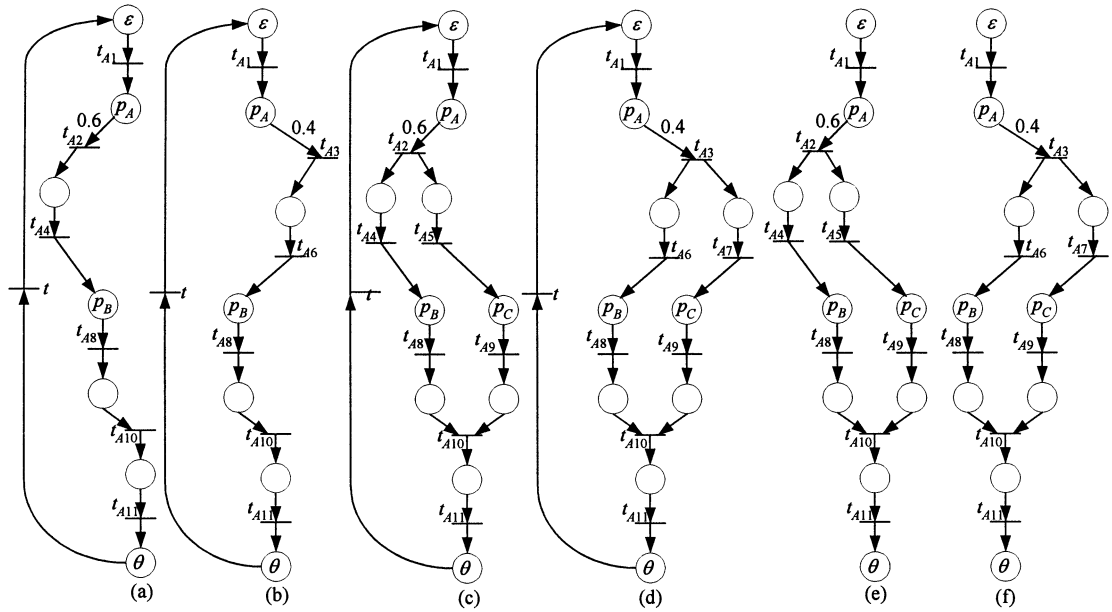

Fig. 8. Decomposition process of w.

in these two transition paths, Step 4 ends without entering its sub-steps. Step 5 deletes t and the corresponding arcs from the two elements of R, i.e., the subnets shown in Fig. 8(c) and (d). Fig. 8(e) and (f) show the two resulting FC-subnets of the model decomposition of w_1.

The model decomposition process of w_2 is shown in Fig. 9. The circuit in Fig. 9(a) is built in Step 2, in which $p^1 = p_J$. After Step 3, a new subnet shown in Fig. 9(b) is added to R. In Step 4, transition path $path = \{t_{B1}, p_D, t_{B2}, t_{B3}, p_G, t_{B4}\}$ is merged to the two elements in R, respectively. Then, two subnets in Fig. 9(c) and (d) are obtained. However, there is a choice place p_G in *path*. Then, it is necessary now to enter the sub-steps of Step 4. Initially, R'_1 includes two subnets shown in Fig. 9(c) and (d) and $R = \phi$. After Steps 4.1–4.3, $PL' = \{p_G\}$. In Step 4.4, only one place path $\{p_G, t_{B5}, p_F, t_{B7}, p_E, t_{B9}, p_D\}$ is constructed. Since p_D belongs to a path from ε to p_G, the place path must be a part of an ICS. Then, it is merged to the original subnets of Fig. 9(c) and (d) and two new subnets, which replace the original ones of R'_1, are obtained as in Fig. 9(e) and (i). Since there are new choice places p_E and p_F in both elements of R'_1, it should re-enter sub-steps of Step 4. These two places are in the same path, and two iterations are needed. In the first iteration, p_F is selected as a proper choice place, and two new propagated subnets (using $\{p_F, t_{B6}, p_E\}$ to replace $\{p_F, t_{B7}, p_E\}$) shown

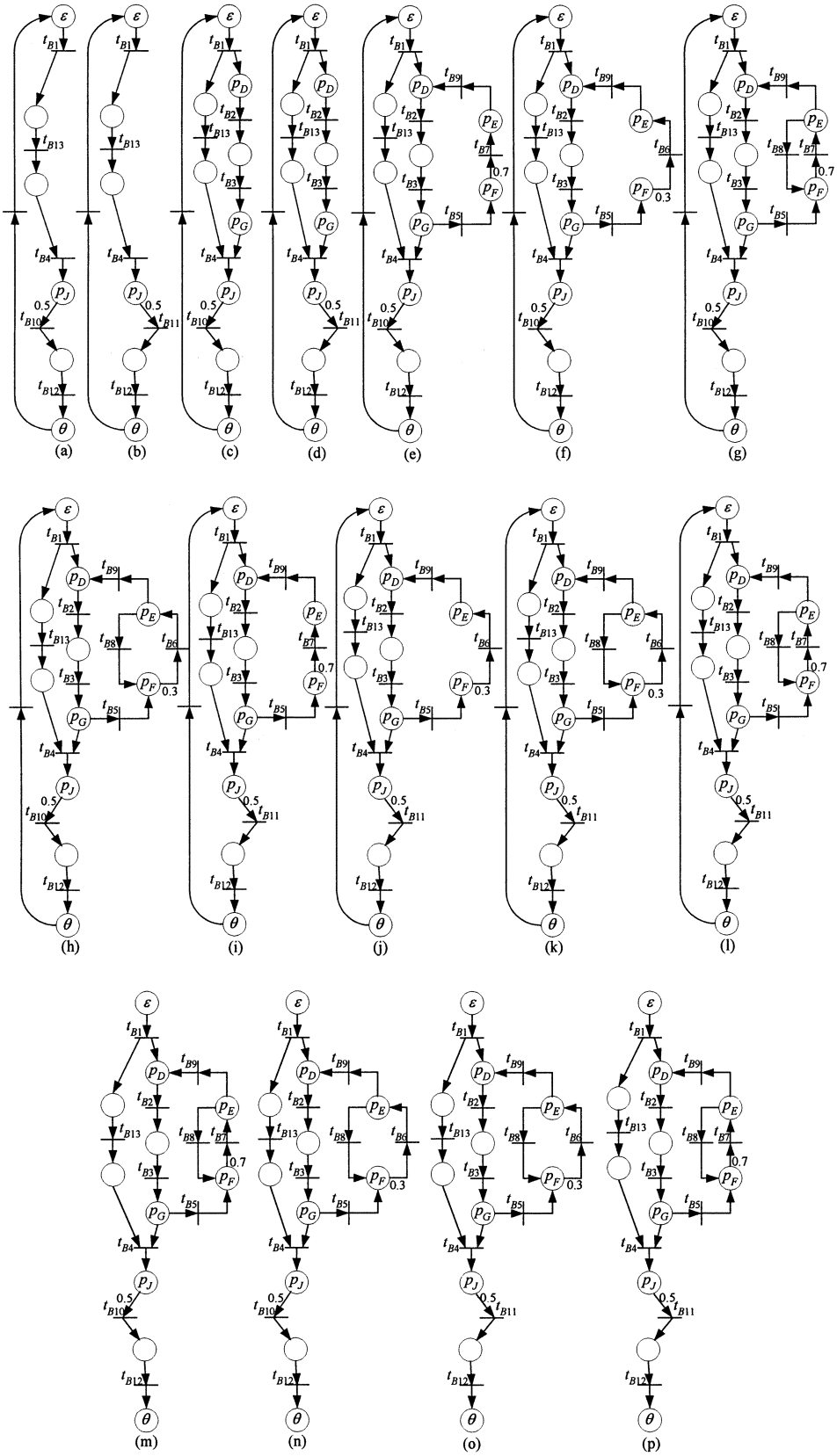

Fig. 9. Decomposition process of w_2.

in Fig. 9(f) and (j) are added to R'_1. Now, there are four subnets shown respectively in Fig. 9(e-f)–(i-j) in R'_1. In the second iteration, p_E is selected as a proper choice place, and place path $\{p_E, t_{B8}, p_F\}$ belonging to a part of an ICS is merged to all the four subnets in R'_1. Now, merge R'_1 into R and Step 4 stops.

There are four subnets shown respectively in Fig. 9(g), (h), (k), and (l) in R. Finally, the resulting FC-subnet sets of w_2, which includes four elements shown in Fig. 9(m)–(p), are obtained after Step 5.

Now, we know w_1 is built for the processing of two types of transaction instances I_1 and I_2, and the FC-subnets shown in Fig. 8(e) and (f) specify their routing paths in w_1, respectively. w_2 is built for the processing of 4 types of transaction instances I_3, I_4, I_5, and I_6, and the FC-subnets shown in Fig. 9(m)–(p) specify their routing paths in w_2, respectively. As mentioned above, the percentage of each type transaction instance account for the total transaction instance processed by corresponding TWF-net is the product of appointed probability of all the choice place's output branch belonging to ACS in its corresponding FC-subnet. From Figs. 8(e) and (f) and 9(m)–(p), we know $\beta^1 = [\beta_1^1, \beta_2^1] = [0.6, 0.4]$ for w_1 and $\beta^2 = [\beta_1^2, \beta_2^2, \beta_3^2, \beta_4^2] = [0.35, 0.15, 0.15, 0.35]$ for w_2. Assuming that the total transaction instance arrival rates of these two TWF-nets are $\lambda^1 = 0.1$ and $\lambda^2 = 0.2$, we can calculate $\lambda_I^1 = [0.06, 0.04]$ and $\lambda_I^2 = [0.07, 0.03, 0.03, 0.07]$, where the jth item of λ_I^I (j = 1,2 for $i = 1$, and j = 1,2,3,4 for $i = 2$) represents the corresponding type transaction instance's arrival rate of the jth FC-subnet of w_i.

Using the relative iterative coefficient computing method, we can acquire $\Omega(t)$ of all the transitions relative to their corresponding FC-subnets. Then, the transition *vs*. FC-subnet matrices for $w_k (k = 1, 2)$ can be constructed as the equation at the bottom of the page, where each item b_{ij} ($1 \le i \le 2, 1 \le j \le 11$ for B_1, and $1 \le i \le 4$, $1 \le j \le 13$ for B_2) in $B_1(B_2)$ represents the average firing times of transition $t_{Aj}(t_{Bj})$ to complete the processing of $I_i(I_{i+2})$ type transition instance. Now, the average transaction instance arrival rate of each transition in $w_1(w_2)$ is obtained by calculating

$$\lambda_{T_1} = \lambda_I^1 * B_1 = [0.1, 0.06, 0.04, 0.06, 0.06, 0.04, 0.04, 0.1, 0.1, 0.1, 0.1]$$
$$(\lambda_{T_2} = \lambda_I^2 * B_2 = [0.2, 0.25, 0.25, 0.2, 0.05, 0.05, 0.05, 0.05, 0.05, 0.1, 0.1, 0.2, 0.2]),$$

where each item of $\lambda_{T_1}(\lambda_{T_2})$ represents the average arrival rate of the corresponding transition in $w_1(w_2)$. Then, $\rho_t = [\lambda_{t_1}/\mu_{t_{A1}}, \ldots, \lambda_{t_A11}/\mu_{t_11}, \lambda_{t_{B1}}/\mu_{t_{B1}}, \ldots, \lambda_{t_{B13}}/\mu_{t_{B13}}=$ $[0.7, 0.36, 0.12, 0.24, 0.3, 0.12, 0.04, 0.8, 0.6, 0.6, 0.3, 0.4, 1.5, 2, 0.2, 0.1, 0.1, 0.15, 0.2, 0.05, 0.1, 0.6, 0.8, 1]=$ $[0.7, 0.36, 0.12, 0.24, 0.3, 0.12, 0.04, 0.8, 0.6, 0.6, 0.3, 0.4, 1.5, 2, 0.2, 0.1, 0.1, 0.15, 0.2, 0.05, 0.1, 0.6, 0.8, 1]$.

The information about mapping relations between the three perspectives of the MWF-net, i.e., F_P and F_R, are specified in the transition *vs* role and role vs. resource matrices

$$C_1 = \begin{bmatrix} 0, & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 1 & 0 \\ 1, & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0, & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0, & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0, & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 \end{bmatrix}$$

$$C_2 = \begin{bmatrix} 0, & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 \\ 0, & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 1 & 0 \\ 1, & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0, & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 \\ 0, & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

$$D = \begin{bmatrix} 0.5, & 0 & 0.5 & 0 & 0 & 0 \\ 0, & 0.4 & 0 & 0.4 & 0 & 0.2 \\ 0.2 & 0 & 0.8 & 0 & 0 & 0 \\ 0, & 0.5 & 0 & 0 & 0.5 & 0 \\ 0, & 0 & 0 & 0.4 & 0 & 0.6 \end{bmatrix}.$$

If firing transition $t_{Aj}(t_{Bk})$, where $1 \le j \le 11 (1 \le k \le 13)$, needs the support of role $o_i (1 \le i \le 5)$, $c_{ij}(c_{ik})$ in $C_1(C_2)$ are set to 1; and otherwise 0. Each item $d_{ij} (1 \le i \le 5, 1 \le j \le 6)$ denotes the probability of the service requests (transaction instance processing) of role o_i bing allocated to r_j. Then, matrix $C = [C_1|C_2]$ can be constructed, and $A_{24}\times_6 = C^T * D$, where each item $a_{ij} (1 \le i \le 24, 1 \le j \le 6)$ represents the probability of the arrival transaction instances (service request) of transition t_i being dispatched (through the perspective mapping) to resource pool r_j. Now, we have $\boldsymbol{H} = [h_1, h_2, \ldots, h_6] = \rho_t * A_{24\times 6} = [2.593, 1.488, 4.657, 1.284, .46, 0.898]$. The workload of all the resource pools can be obtained and $\rho_r = [\rho_{r_1}, \rho_{r_2}, \ldots, \rho_{r_6}] = [0.8995, 0.7890, 0.5908, 0.3307, 0.2439, 0.4792]$. The resource pool r_1 is the potential bottleneck of the MWF-net since $\rho_{R_1} > \rho_{R_i}, i \ne 1$.

To obtain the lower bound of transition instance I_i's $(1 \le i \le 6)$ turnaround time, we should use Theorems 1-2 iteratively to reduce the I_i's corresponding FC-subnet. Next, we use the FC-subnet in Fig. 9(m), which specifies the routing path of transaction instance I_3 in w_2, to illustrate how to acquire the lower bound of average turnaround time of I_3. Applying Theorem 2 to the ICS which includes circuit $\{p_E, t_{B8}, p_F, t_{B7}, p_E\}$ in the FC-subnet of Fig. 9(m), the subnet shown in Fig. 10(a) is obtained, where $1/\mu_{t'_{B7}} = 6$ and $1/\mu_{t'_{B8}} = 4$. Then, we reduce SCS $\{p_G, t_{B5}, p_F, t'_{B7}, p_E, t'_{B8}, t_{B9}, P_D\}$ in Fig. 10(a) to the $\{p_c, t_{B5678}, p_D\}$ in Fig. 10(b), where $1/\mu_{t_{B5789}} = 13$. Similarly, we transform the subnet in Fig. 10(b) into Fig. 10(c),

$$B_1 = \begin{bmatrix} 1, & 1 & 0 & 1 & 1 & 0 & 0 & 1 & 1 & 1 & 1 \\ 1, & 0 & 1 & 0 & 0 & 1 & 1 & 1 & 1 & 1 & 1 \end{bmatrix} \begin{matrix} I_1 \\ I_2 \end{matrix} \text{ and}$$

$$B_2 = \begin{bmatrix} 1, & 1.25 & 1.25 & 1 & 0.25 & 0 & 0.5 & 0.25 & 0.25 & 1 & 0 & 1 & 1 \\ 1, & 1.25 & 1.25 & 1 & 0.25 & 0.5 & 0 & 0.25 & 0.25 & 1 & 0 & 1 & 1 \\ 1, & 1.25 & 1.25 & 1 & 0.25 & 0 & 0.5 & 0.25 & 0.25 & 0 & 1 & 1 & 1 \\ 1, & 1.25 & 1.25 & 1 & 0.25 & 0.5 & 0 & 0.25 & 0.25 & 0 & 1 & 1 & 1 \end{bmatrix} \begin{matrix} I_3 \\ I_4 \\ I_5 \\ I_6 \end{matrix}$$

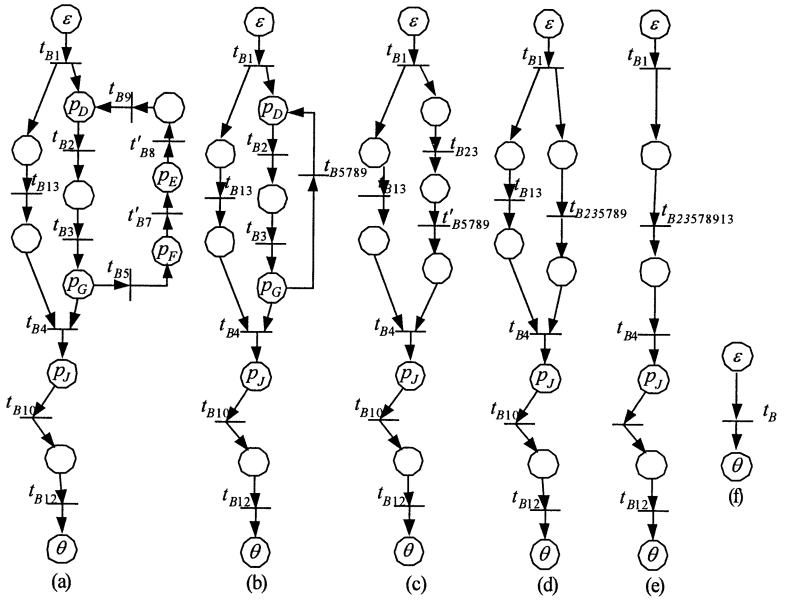

Fig. 10. Transaction instance I_3's average turnaround time lower bound computing.

and then Fig. 10(d). The equivalent transition $t_{B2\,356\,789}$'s average firing delay is $1/\mu_{t_{B235\,789}} = 20.75$. Then, we transform two branches in CCS in Fig. 10(d) to $t_{B23\,578\,913}$ in Fig. 10(e), whose average firing delay is 21.72. Finally, the subnet in Fig. 10(e) is simplified as to Fig. 10(f), where the equivalent transition t_B's average firing delay is 29.72. Hence, the lower bound of I_3's average turnaround time in the MWF-net is 29.72. Similarly, the lower bounds of transaction instances I_1, I_2, I_4, I_5, and I_6 are 39.26, 32.72, 29.24, 34.24, and 34.72, respectively. During the structure transforming, Erlang, gamma, hypo and hyper-exponential distributions are all approximated as exponential distribution. However, since they have similar properties [33], the error of this computing method of the turnaround time' lower bound is small. This can be seen from the simulation.

To verify the analysis results of the MWF-net, we transform the MWF-net into a directed-activity-network based workflow model. Then the performance experiments are done with the CIMFlow-Simulator [1], [36], which is a workflow simulation tool supporting business process reengineering. In CIMFlow-Simulator, there are two kinds of transaction instance generators, i.e., the inter-arrival and completion generators, which are fit for the experiments of workload and lower bounds of different types transaction instance's average turnaround time, respectively. Inter-arrival generator issues transaction instances at a pre-defined rate, which are used in the experiment of workload. For the six resource pools, the simulation results of their workloads are 0.8542, 0.7500, 0.5612, 0.3018, 0.2420, and 0.4500, respectively. Obviously, the simulation results are a little smaller than the analysis results. The reason is that the information about resource failure and repair is not considered in the workflow simulation model since the current simulation tool does not have the capability to simulate resource failure and repair functions. Completion generator introduces a transaction instance into the process, and then introduces another transaction only after the previous one has completed its journey through the process. Since at anytime there is only one transaction instance in a workflow process (corresponding to a TWF-net), there is no resource contention. Then it is used in the experiment of lower bounds of different type transaction instance' average turnaround time. The simulation results of the lower bounds of 6 transaction instances' average turnaround time are 37.6845, 35.3824, 28.8628, 27.5600, 33.9152, and 32.0810, respectively. These results are close to the analysis results.

VII. Conclusion

Studies on workflow performance-related issues are receiving more and more attentions. All the routing constructs of a workflow are mapped into a high-level stochastic PN (SPN), and then throughput time of the process is analytically computed [19]. Based on four performance equivalent formulae, an approximate performance analysis method of a workflow is presented in [20]. These two techniques both assume the infinite availability of resources in the workflow configuration, and cannot be applied to a workflow process with complex control structures (e.g., the interlacing CCS and ACS in the TWF-net of Fig. 7(a)). Generalized stochastic PN (GSPN) is used to model workflow [21], [23], and then a method based on a CTMC is used to obtain lower bounds of the execution performance. A simple GSPN, which is a so-called *load equivalence aggregation* model, has been developed in [22], and then the model is simulated using a Colored GSPN to obtain some performance related measures of human resources in a workflow. Based on the definition of *change time*, the performance evaluation for the dynamic workflow changes can be conducted using the method proposed in [17]. However, the technique can be used for only *acyclic* time WF-net whose transaction instance arrival interval is constant. The workflow model is modeled as a queuing network [18], [38]. The method for identifying the critical path of a *nested workflow* model and determining the minimum number of servers for the critical

activity is derived [18]. Some approximation approaches are employed in [38] for workforce configuration, and then bound the analysis is conducted for the corresponding network. But these techniques are not immediately applicable since they both assume that dedicated servers exist for an activity's execution.

Existing modeling and analysis techniques used to resolve different aspects of workflow performance-related problems have mainly two shortcomings, which restrict their application in practice. One is that only one isolated workflow process is considered. In general, however, there are several workflow processes (logistics of business process) running concurrently in the enactment environment. The other is that almost no organization and resource information is considered. Although some literatures incorporate resource information into their workflow models, only some simple disciplines, such as dedicated server for an activity's execution [17], [18] or infinite servers semantics [19], [20] are supposed. They are not necessarily true in actual workflow systems.

This paper for the first time considers all the necessary information for the performance related theoretical analysis of a workflow model. First, the TWF-net is obtained by extending WF-net with time information. Then, to provide a formal framework for workflow model performance analysis, the concept of MWF-net that includes multiple TWF-nets, as well as the organization and resource information is given. Different from the classical timed and/or stochastic PN, in which resource configuration and allocation information is modeled in the net structure by additional resource places and corresponding directed arcs, thus increasing drastically the complexity of the net structure and destroying the free choice characteristic of a sound TWF-net, the TWF-nets in the MWF-net only specify the control logic. The additional information about resource, organization and perspectives mapping relations is used to specify the static and dynamic resource allocation. Since at anytime there is only one transaction instance in a workflow process (corresponding to a TWF-net), there is no resource contention. Then it is used in the experiment of lower bounds of different type transaction instance' average turnaround time. The simulation results of the lower bounds of six transaction instances' average turnaround time are close to the analysis results.

To realize the performance analysis of the MWF-net, we present a decomposition algorithm from a free-choice TWF-net to a set of FC-subnets, each of which describes the routing path of a type of transaction instances. Based on the resource availability analysis, relative iterative coefficient computing for each transition in the corresponding FC-subnet, and the perspectives mapping relation analysis, the routing of transaction instances in the TWF-net is mapped into the service request arrival rate of the resource pools. Then, the workload of each resource pool can be obtained. In addition, since there are only ICSs, SCSs, and CCSs in each element of the resulting FC-subnet set, two theorems are introduced for the turnaround times' lower bound computing for all types of transaction instances processed in the MWF-net. The case study shows that our method can be effectively utilized in practice.

Although the performance related analysis method has exponential complexity, the complexity grows exponentially only with the number of choice places (not all the places) in the TWF-net and linearly with the number of TWF-nets in the MWF-net.

During the discussion of workflow performance related analysis, we implicitly assume that any two TWF-nets in the MWF-net do not share transitions. The techniques proposed in this paper need to be extended to deal with the case of multiple TWF-nets share one or more transitions. This will be left for future exploration.

REFERENCES

[1] Y. S. Fan, *Fundamentals of Workflow Management Technology. Tsinghua*. New York: Springer-Verlag, 2001.

[2] "Workflow Management Coalition. Interface 1:Process Defition Interchange, Process Model,", Doc. No. WfMC TC-1016-P, 1998.

[3] S. Jablonski and C. Bussler, "Workflow management: Modeling concepts," in *Architecture, and Implementation* . London, U.K.: Int. Thomson Computer Press, 1996.

[4] W. M. P. van der Aalst and A. H. M. ter Hofstede, "Verification of workflow task structures: A petri-net-based approach," *Inf. Syst.*, vol. 25, no. 1, pp. 43–69, 2000.

[5] W. Sadiq and M. E. Orlowska, "Analyzing process models using graph reduction techniques," *Inf. Syst.*, vol. 25, no. 2, pp. 117–134, 2000.

[6] N. R. Adam, V. Atluri, and W. K. Huang, "Modeling and analysis of workflow using petri nets," *J. Intell. Inform. Syst.*, vol. 10, no. 2, pp. 1–29, 1998.

[7] W. M. P. Van Der Aalst, "The application of petri nets to workflow management," *J. Circuits, Syst. Comput.*, vol. 8, no. 1, pp. 21–66, 1998.

[8] Y. Han, J. Himmighofer, T. Schaaf, and D. Wikarski, "Management of workflow resource to support runtime adaptability and system evolution," in *Proc. 1st Int. Conf. Practical Aspects Knowledge Management* , Basel, Switzerland, Oct. 1996.

[9] J. Hu, "Research on Adaptive Workflow Management System to Support Dynamical BPR," Ph.D. dissertation, Computer Integrated Technol. Lab., Shanghai Jiaotong Univ., Shanghai, China, 2001.

[10] E. Panagos *et al.*, "Reducing escalation-related costs in WFMSS," in *Proc. NATO Advanced Study Institute Workflow Management Systems Interoperability*, A. Dogac *et al.*, Eds., Istanbul, Turkey, August 1997.

[11] H. Pozewauning, J. Eder, and W. Liebhart, "EPERT: Extending PERT for workflow management systems," in First East Europ. Symp. Advances Database Information Systems, St. Petersburg, Russia, Sept. 1997.

[12] Y. Qu, C. Lin, and J. Y. Wang, "Linear Temporal Inference of Workflow Management systems Based on Timed Petri Nets Models," in *EDICS 2002*, Y. Han, S. Tai, and D. Wikarski, Eds. Berlin, Germany: Springer-Verlag, 2002, vol. 2480, Lecture Notes Computer Science, pp. 30–44.

[13] H. Zhuge, T.-Y. Cheung, and H.-K. Pung, "A timed workflow process model," *J. Syst. Softw.*, vol. 55, pp. 231–243, 2001.

[14] J. Q. Li, Y. S. Fan, and M. C. Zhou, "Timing constraint workflow nets for workflow analysis," IEEE Trans. Syst., Man, Cybern. A, vol. 33, Mar. 2003, to be published.

[15] J. Eder, E. Panagos, H. Pozewaunig, and M. Rabinovich, "Time management in workflow systems," in *Proc. Int. Conf. Business Information Systems*, Apr. 1999, pp. 266–280.

[16] J. Eder, E. Panagos, and M. Rabinovich, "Time constraints in workflow systems," in *Advanced Information System Engineering*. Berlin, Germany: Springer-Verlag, 1999, vol. 1626, Lecture Notes in Computer Science, pp. 286–300.

[17] S. Yamaguchi, G. Qi-Wei, and M. Tanaka, "Performance evaluation on change time of dynamic workflow changes. IEICE transactions on fundamentals of electronics," *Commun. Comput. Sci.*, vol. E83-A, no. 11, pp. 2177–2187, 2000.

[18] J. H. Son and M. H. Kim, "Improving the performance of time-constrained workflow processing," *J. Syst. Softw.*, vol. 53, pp. 211–219, 2001.

[19] K. M. van Hee *et al.*, "Using formal analysis techniques in business process redesign," in *Business Process Management*, W. van der Aalst *et al.*, Eds. Berlin, Germany: Springer-Verlag, 2000, vol. 1806, Lecture Notes Computer Science, pp. 142–160.

[20] C. Lin, Y. Qu, F. Y. Ren, and D. C. Marinescu, "Performance equivalent analysis of workflow systems based on stochastic petri net models," , Lecture Notes Computer Science #2480, 2002, pp. 64–79.

[21] A. Ferscha, "Qualitative and quantitative analysis of business workflows using generalized stochastic petri nets," in *Proc. Workflow Management-Challenges, Paradigms Products*, G. Chroust and A. Benczur, Eds., Linz, Austria, Oct. 19–21, 1994, pp. 222–234.

[22] A. K. Schomig and H. Rau, "A Petri Net Approach for the Performance Analysis of Business Process," Univ. Wurzburg, Schloss Dagstuhl, Germany, Rep. 116 Semiar at IBFI, 1995.
[23] A. Ferscha, "Business workflow analysis using generalized stochastic petri nets," in *Proc. 9th Austrian-Hungarian Informatics Conf.*, 1994, pp. 222–234.
[24] M. Gillmann, J. Weissenfels, G. Weikum, and A. Kraiss, "Performance and Availability Assessment for the Configuration of Distributed Workflow Management Systems," presented at the Conf. Extending Database Technology, Konstanz, Germany, 2000.
[25] K. H. Kim and C. A. Ellis, "Performance analytic models and analyzes for workflow architectures," *Inf. Syst. Frontiers*, vol. 3, no. 3, pp. 339–355, 2001.
[26] K. Salimifard and M. Wright, "Petri net-based modeling of workflow systems: An overview," *Eur. J. Oper. Res.*, vol. 134, pp. 664–676, 2001.
[27] W. M. P. van der Aalst *et al.*, "Chapter 10: Three good reasons for using a petri-net-based workflow management system," in *Information and Process Integration in Enterprise: Rethinking docmuments*, T. Wakayama *et al.*, Eds. Norwell, MA: Kluwer, 1998, pp. 161–182.
[28] M. D. Zisman, "Representation, Specification and Automation of Office Procedures," Ph.D. dissertation, Univ. Pennsylvania, Philadelphia, Wharton Business, 1977.
[29] D. Georgakopoulos, M. Hornick, and A. Sheth, "An overview of workflow management: From process modeling to workflow automation infrastructure," *Distrib. Parallel Databases*, vol. 3, pp. 119–153, 1995.
[30] H. P. Lin, Y. S. Fan, and C. Wu, "Workshop scheduling research based on workflow management technology," in *Proc. IE&EM*, Tianjin, China, Aug. 10–12, 2001, pp. 252–256.
[31] M. Petri, "Nets: Properties, analysis and applications," *Proc. IEEE*, vol. 779, no. 4, pp. 541–580, Apr. 1989.
[32] M. A. Marsan, G. Balbo, A. Bobbio, G. Chiola, G. Conte, and A. Cumani, "The effect of execution policies on the semantics and analysis of stochastic petri nets," *IEEE Trans. Softw. Eng.*, vol. 15, pp. 832–846, July 1989.
[33] K. S. Trivedi, *Probability & Statistics With Reliability, Queuing, and Computer Science Applications*. Englewood Cliffs, NJ: Prentice-Hall, 1982.
[34] H. C. Tijms, *Stochastic Models*. New York: Wiley, 1994.
[35] D. Gross and C. M. Harris, *Fundumental of Queuing Theory*, 3rd ed. New York: Wiley, 1998.
[36] Business Process Modeling and Workflow Technology Forum. [Online]. Available: http://www.simflow.net
[37] M. C. Zhou and F. DiCesare, *Petri Net Synthesis for Discrete Event Control of Manufacturing Systems*. Boston, MA: Kluwer, 1993.
[38] Y. Kevin, K. Ng, A. Ghanmi, M. N. Lam, and R. E. Mitchell, "Workforce configuration and workflow analysis of an information technology organization: A queueing network approach," *IEEE Trans, Syst., Man, Cybern. A*, vol. 32, pp. 724–733, Nov. 2002.

JianQiang Li received the B.S. degree in Mechatronics from Beijing Institute of Technology, Beijing, China, in 1996. He is currently pursuing the Ph.D. degree in control theory and control engineering at Tsinghua University, Beijing, China.

His research interest includes workflow modeling and optimization, business process reengineering, cooperated complex systems, and system integration.

YuShun Fan received the B.S. degree in automatic control from Beijing University of Aeronautics and Astronautics, Beijing, China, in 1984, and the M.S. and Ph.D. degrees in control theory and application from Tsinghua University, Beijing, in 1987 and 1990, respectively.

He is currently Professor of the Department of Automation, Vice Director of the System Integration Institute, and Director of the Networking Manufacturing Laboratory, Tsinghua University. His research interest includes enterprise modeling methods and optimization analysis, business process reengineering, workflow management, system integration and integrated platform, object-oriented technologies and flexible software systems, Petri nets modeling and analysis, workshop management and control. He authored six books in enterprise modeling, workflow technology, intelligent agent, and object oriented complex system analysis, computer integrated manufacturing, respectively, and published more than 140 research papers in journals and conferences. He is a member of the IFAC Advanced Manufacturing Technology Committee and a member of National High-Tech. CIMS Expert Group of China. From September 1993—March 1994, he was a Visiting Scientist at the University Bochum, Germany, supported by Federal Ministry for Research and Technology. From April 1994 to July 1995, he was a Visiting Scientist, supported by Alexander von Humboldt Stiftung, at Fraunhofer Institute for Production System and Design Technology (FhG/IPK), Germany.

Dr. Fan served on the Program Committees of the 1992 International Symposium on CIM, Beijing, China, 1997 IEEE International Conference on Factory Automation and Emerging Technology, Los Angeles, CA, and 2002 International Workshop on Emergent Technologies in Engineering Cooperative Information Systems, Beijing, China.

MengChu Zhou (S'88–M'90–SM'93–F'03) received the B.S. degree from Nanjing University of Science and Technology, Nanjing, China, in 1983, the M.S. degree from Beijing Institute of Technology, Beijing, China, in 1986, and the Ph.D. degree in computer and systems engineering from Rensselaer Polytechnic Institute, Troy, NY, in 1990.

He joined New Jersey Institute of Technology (NJIT), Newark, in 1990, where he is currently Professor of electrical and computer engineering and the Director of Discrete-Event Systems Laboratory. His research interests are in computer-integrated systems, Petri nets, semiconductor manufacturing, multilifecycle engineering, *ad hoc* networks, and system security. He has over 200 publications including four books, over 50 journal papers, and ten book chapters. He coauthored, with F. DiCesare, *Petri Net Synthesis for Discrete Event Control of Manufacturing Systems* (Norwell, MA: Kluwer, 1993), edited *Petri Nets in Flexible and Agile Automation* (Norwell, MA: Kluwer, 1995), and coauthored, with K. Venkatesh, *Modeling, Simulation, and Control of Flexible Manufacturing Systems: A Petri Net Approach* (Singapore: World Scientific, 1998). He has been an editor of *International Journal of Intelligent Control and Systems* since 1996.

Dr. Zhou has been invited to lecture in Australia, Canada, China, France, Germany, Hong Kong, Italy, Japan, Korea, Mexico, Taiwan, and the United States. He served as Associate Editor of IEEE TRANSACTIONS ON ROBOTICS AND AUTOMATION from 1997 to 2000 and is currently Associate Editor of the IEEE TRANSACTIONS ON SYSTEMS, MAN, AND CYBERNETICS and the IEEE TRANSACTIONS ON AUTOMATION SCIENCE AND ENGINEERING. He organized and chaired over 60 technical sessions and served on program committees for many conferences. He was Program Chair of the 1998 and Co-Chair of the 2001 IEEE International Conferences on System, Man, and Cybernetics (SMC) and the 1997 IEEE International Conference on Emerging Technologies and Factory Automation and Guest Editor for the IEEE TRANSACTIONS ON INDUSTRIAL ELECTRONICS and the IEEE TRANSACTIONS ON SEMICONDUCTOR MANUFACTURING. He was General Co-Chair of the 2003 IEEE International Conference on Systems, Man, and Cybernetics, Washington, DC, October 5–8, 2003. He is serving as a founding Co-General Chair of the 2004 IEEE International Conference on Networking, Sensing, and Control. He has led or participated in 26 research and education projects funded by the National Science Foundation, Department of Defense, Engineering Foundation, New Jersey Science and Technology Commission, and industry. He was the recipient of the National Science Foundation's Research Initiation Award, the CIM University-LEAD Award of the Society of Manufacturing Engineers, the Perlis Research Award from NJIT, the Humboldt Research Award of the U.S. Senior Scientists, the Leadership Award and Academic Achievement Award of the Chinese Association for Science and Technology-USA, and the Asian American Achievement Award of the Asian American Heritage Council of New Jersey. He is Chair (founding) of the Discrete Event Systems Technical Committee of the IEEE Systems, Man, and Cybernetics Society and Co-Chair (founding) of the Semiconductor Factory Automation Technical Committee of the IEEE Robotics and Automation Society. He is a life member of the Chinese Association for Science and Technology-USA and served as its President in 1999.