E-Net Modeling and Analysis of Emergency Response Processes Constrained by Resources and Uncertain Durations

Cong Liu, Qingtian Zeng, Hua Duan, MengChu Zhou, *Fellow, IEEE*, Faming Lu, and Jiujun Cheng

***Abstract*—Time and resource management and optimization are two important challenges for an emergency response process, by which all individuals and groups manage hazards in an effort to avoid or ameliorate the impact of disasters. Compared with a traditional business process, an emergency response process has its own features. To our best knowledge, there is no formal method to model and analyze emergency response processes by taking uncertain activity execution duration, resource quantity, and resource preparation duration into account. This paper presents such a method based on an E-Net that is a Petri net-based formal model for an emergency response process constrained by resources and uncertain durations. According to the number of available resources, execution of an E-Net is classified into the worst, delayed, and best cases. Based on a priority-activity-first strategy and corresponding algorithms, this paper finds the duration to execute each activity for the delayed case. By experiments, we prove that the proposed strategy can ensure shorter execution duration of the whole process than a conventional one. A running case of a chlorine tank explosion is given to validate the proposed method.**

***Index Terms*—Discrete event systems, emergency response process, Petri nets, resource allocation, timed Petri nets, uncertain duration, workflow management.**

Manuscript received June 14, 2013; revised March 17, 2014; accepted May 28, 2014. Date of publication July 10, 2014; date of current version December 12, 2014. This work was supported in part by the National Science Foundation of China under Grant 61170079 and 61202152, in part by the International S&T Cooperation Program of China under Grant 2013DFM10100, in part by the National Science Foundation under Grant CMMI-1162482, in part by Natural Science Foundation Programs of Shanghai under Grant 13ZR1443100, in part by the Science and Technology Development Fund of Qingdao under Grant 13-1-4-153-jch, and in part by the open project of the Key Laboratory of Embedded System and Service Computing, Ministry of Education, Tongji University under Grant ESSCKF201403. This paper was recommended by Associate Editor L. Fang.
(Corresponding author: Jiujun Cheng.)

C. Liu, Q. Zeng, and H. Duan are with the Shandong University of Science and Technology, Qingdao 266590, China (e-mail: liucongchina@163.com; qtzeng@sdust.edu.cn; hduan@sdust.edu.cn).

M. C. Zhou is with The Key Laboratory of Embedded System and Service Computing, Ministry of Education, Tongji University, Shanghai 201804, China and also with the Department of Electrical and Computer Engineering, New Jersey Institute of Technology, Newark, NJ 07102 USA (e-mail: zhou@njit.edu).

F. Lu is with the Key Laboratory of Embedded System and Service Computing, Tongji University, Shanghai 200092, China, and also with the Shandong University of Science and Technology, Qingdao 266590, China (e-mail: fm_lu@163.com).

J. Cheng is with the Key Laboratory of Embedded System and Service Computing, Ministry of Education, Tongji University, Shanghai 200092, China (e-mail: chengjj@tongji.edu.cn).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TSMC.2014.2330555

I. Introduction

Emergency response systems (ERSs) resulted from the need for a new method to manage rapidly moving wildfires in the early 1970s. At that time, emergency managers faced a number of problems, such as too many people reporting to one supervisor, lack of reliable and accurate incident information, and inadequate and incompatible communications. They needed effective emergency response management systems (ERMSs) badly. Since the 9/11 terrorist attack event, there have been considerable efforts to improve the ability to respond to emergencies. Thus, research on ERMSs has been drastically intensified [1]–[12]. Similar to the workflow management systems (WFMSs) [13]–[22], ERMSs consist of two main functional components: emergency modeling and emergency enactment. Most of the existing work pays more attention to the former by focusing on time performance analysis [1]–[6] and resource scheduling [7]–[12].

An emergency is a situation that poses an immediate risk to human health, life, and property, which requires urgent interventions to prevent its worsening. These interventions are organized as a process that is usually described in an emergency plan, named an emergency response process. In this process, all individuals, groups, and communities manage hazards in an effort to avoid or reduce the impact of disasters. It is based on the idea that an emergency response process is quite similar to a business process and, therefore, can be modeled as a workflow. A workflow is a representation of a given process that is made up of well-defined activities, also referred to as tasks. Modeling and analysis of workflows have been studied for several decades. A large number of researchers have done much work in the area of resource conflict resolution [13]–[15], structural correctness verification [16]–[18], and time performance evaluation [19]–[21]. As a tool to model and analyze discrete event systems, Petri nets [24]–[28] have shown their great power in dealing with concurrence and conflicts. Because of this advantage, they have been widely used to model, analyze, control, and verify flexible manufacturing systems [23], WFMSs [13]–[22], and other discrete event systems [29]–[33]. Main reasons to use them to model and analyze workflows include [34]: 1) their graphical nature and formal semantics; 2) their explicit model of a case state; and 3) the availability of many analysis techniques for them.

The existing workflow modeling and analysis techniques cannot be applied directly to emergency response processes because they have some requirements different from those of the traditional workflows.

1) They exhibit considerable flexibility and uncertainty. An emergency response process must be able to grow or shrink to cope with frequent changes caused by incoming events. Therefore, during the modeling stage, it is impossible to estimate the real duration of each activity before its execution.
2) They exhibit close relations between activities and resources. In an ERMS, large quantities of resources, including emergency responders, ambulances, fire trucks, medicines, food, and clothing, are required. The lack of resources can cause the contention, need for some activities to wait for others to complete, and delayed accomplishment of a whole process. To cope with this problem, resources should be modeled with more accuracy. Thus, we divide resources into reusable and consumable ones. Both need time to be prepared for deployment. This duration is also of uncertain nature.
3) An emergency can cause a fire if flammable substance is involved, an explosion if propelling substance is involved, and a toxic cloud if toxic and gaseous substance is involved. Risk of human health, property loss, and environmental damage can be high, which are rare in a traditional workflow. Thus, effective ERMSs are highly demanded for the successful completion of an emergency mission.
4) An emergency response process is a real-time service, where timeliness is critical to mission success. Thus, optimization is needed to ensure its quality completion.

In this paper, by taking these four requirements of emergency response processes into account, we first introduce a Petri net-based formal model for an emergency response process constrained by resources and uncertain durations, called E-Net for short. Then, we discuss the modeling and analysis approaches for emergency response processes. Without considering resource constraints, we obtain ideal execution duration for an E-Net. According to the number of available resources, execution cases of an E-Net are classified. Conflict detection and resolution strategies are proposed to optimize its real execution.

The remainder of this paper is organized as follows. Section II discusses the related work. In Section III, a formal specification for an emergency response process constrained by resources and uncertain durations is first introduced. Then, the scenario of a chlorine tank explosion is presented. Finally, the formal definition of E-Nets is given. Section IV discusses modeling approaches for emergency response processes based on E-Nets. Section V discusses resource management and three execution cases of E-Nets. In Section VI, the real execution duration for the delayed case is analyzed. A priority-activity-first strategy is proposed to resolve resource conflicts, and experimental evaluation is conducted. Section VII gives a running example to show the proposed approaches. Finally, Section VIII draws the conclusion.

II. Related Work

We summarize the existing work related to time and resource management of emergency response processes.

A. Time Management

Kang [1] presents an assessment of emergency evacuation in an underground rail station subject to a mid-platform train fire. The results demonstrate that the appropriate incorporation of fire and smoke effects is important in station evacuation analysis. Wang [2] proposes a workflow intuitive and formal approach formalism that takes task execution time into account to support emergency response timeliness analysis. An example of emergency healthcare is used to validate the proposed method. A study on the uncertainty of occupant evacuation time under emergency conditions is presented in [3]. Its stochastic analysis is performed by coupling the uncertainty of fire detection, alarming, and premovement with movement time. The results show that it is a variable influenced by a large number of uncertainties, including emergency evolution dynamics, human behaviors under emergency conditions, and environments. Chen and Tang [4], by considering the probability of a road being destroyed as an important element when roads are not stable, solved the emergency logistics path selection problem with multiple demand points and multiple service points based on the combined time expectation. In [5], the time consumed by a vehicle to arrive at an emergency center is assumed to be subject to a Gaussian distribution and supply loading time consumption is subject to an exponential distribution. The average vehicle waiting-time and waiting queue length expressions are then given. According to this hypothesis, a vehicle arriving and supply loading model is described, and the average vehicle waiting-time and waiting queue length expressions are finally derived. A model for earthquake emergency shelter choices based on constrained optimization is provided in [6]. Its objective is to ensure that the total evacuation time is the shortest by comprehensively considering the choices of evacuation routes.

B. Resource Management

In order to find the best assignment of available resources to operational areas, Fiedrich *et al*. [7] present a dynamic optimization model that uses detailed descriptions of the operational areas and available resources to calculate the resource performance and efficiency. Tsenga *et al*. [8] shows the benefits of developing an adequate emergency response plan with safety and industrial hygiene resources to deal with the effects resulting from a chlorine gas leak, in order to lessen or avoid injury to plant personnel and citizens in the neighboring community. Wang *et al.* [9] pioneer a formal, yet intuitive, approach for the modeling and analysis of emergency response processes, which has taken resources into consideration. Sell and Braun [10] present a model for a WFMS for supporting the modeling, execution, and management of emergency plans before and during a disaster, which supports unstructured activities and the management of resources. The equipment control structure presented in [11] enables decentralized and collective decision-making for equipment prioritization and distribution in response to disasters. In [12], emergency medicine crisis resource management is performed via anesthesia crisis resource management as an example, which can help one determine participant perceptions.

C. Summary

Recently, researches into ERMSs have drawn much public attention and enjoyed an accelerated flush, especially in the area of time and resource management. However, more efforts are required to address their formal modeling, analysis, and optimization issues when facing uncertain activity execution duration, uncertain resource preparation duration, and limited resource quantity.

This paper pays attention to the modeling and analysis methods for an emergency response process constrained by resources and uncertain durations. The number of available resources and minimum resource demand for an emergency response process are analyzed. Resource conflict detection and resolution strategies are also investigated to optimize the system performance.

III. Formal Specification and E-Net

A. Formal Specification of Emergency Response Process

In this section, we first introduce the formal specification of an emergency response process constrained by resources and uncertain durations. It satisfies the following assumptions.

1) Execution duration of each activity is uncertain. Process modelers can only estimate its minimum and maximum execution durations.
2) Each kind of resources needs preparation time before their use. All resources can be started to prepare as soon as the process starts. Meanwhile, resources are divided into reusable and consumable ones.
3) Some activities need resources for their execution. During their execution, if the resources are reusable, they are exclusively used and become available only after they are released. If the resources are consumable, they are consumed during execution and cannot be reused.

The scope of this work is to model and analyze emergency response processes by taking uncertain activity execution duration, resource quantity, and resource preparation duration into account. According to Sun *et al.* [37], a choice structure can be analyzed by its corresponding instances. In other words, we focus our approach on each emergency response process instance only. For an emergency response process with choice structures, the final analysis results can be derived by integrating the results of all its running instances. Moreover, a choice structure usually involves process semantic information [44]. Therefore, an extended E-Net with semantics to describe case attributes and task execution conditions should be further studied.

Let $Z = \{0, 1, 2, \ldots\}$, $Z_n = \{1, 2, \ldots n\}$ where n is a positive integer, and $\mathbb{R}$ be the set of nonnegative real numbers.

Definition 1: An emergency response process constrained by resources and uncertain durations is a nine-tuple $ERP = <Activity, Resource, Time, Rproperty, Relation, f_{AR}, f_{RP}, f_l, f_u>$, where

1) $Activity = \{A_i | i \in Z_n\}$ is an activity set.
2) $Resource = \{r_i | i \in Z_m\}$ is a resource set.
3) $Time = \{time_i | i \in Z_l\}$ is a time duration set, where $time_i \geq 0$.
4) $Rproperty = \{0, 1\}$ is a resource property set, where 0 represents a reusable and 1 a consumable resource.
5) $Relation \subseteq Activity \times Activity$ is a relation set, representing various relations among activities.
6) f_{AR}: $Activity \rightarrow Z^m$ is the resource function of activities. $\forall x \in Activity$, $f_{AR}(x) = (q_1, q_2, \ldots q_m)$, where q_i is the quantity of r_i required by activity x.
7) f_{RP}: $Resource \rightarrow Rproperty$ is a resource property function.
8) Given $x \in Activity \cup Resource$, $f_l(x) \in \mathbb{R}$ is the minimum time required to execute activity x or prepare resource x while $f_u(x) \in \mathbb{R}$ is the maximum one, sastisfying $f_l(x) \leq f_u(x)$.

Definition 1 presents the formal specification of an emergency response process constrained by resources and uncertain durations.

1) *Activity* defines all n activities involved in an emergency response process.
2) *Resource* defines all m required resources, including reusable and consumable ones.
3) Assume $Resource = \{r_1, r_2, r_3\}$ and resource function $f_{AR}(A_i) = <0, 5, 3>$, it means that the execution of A_i requires 5 units of r_2 and 3 units of r_3.
4) Resource property is determined by f_{RP}. By $f_{RP}(r_2) = 0$ and $f_{RP}(r_3) = 1$, we mean that r_2 is reusable while r_3 is consumable.
5) *Relation* defines the connection relations among activities. $\forall A_i, A_j \in Activity$, if $(A_i, A_j) \in Relation$, A_j cannot start before A_i. We call A_i a preactivity of A_j, and A_j as a post-activity of A_i.
6) *Time* defines time constraints for activities and resources. If its actual execution duration of A_i is *Atime*, we have $f_l(A_i) \leq Atime \leq f_u(A_i)$. If its actual preparation duration of r_j is *Rtime*, we have $f_l(r_j) \leq Rtime \leq f_u(r_j)$.

B. Simple Example

Consider a chemical tank explosion scenario. Some of the critical missions in it are to rescue victims and dispose of the leaked chlorine. All activities and resources involved are described as follows.

- A_1: Investigators investigate the injuries.
- A_2: Investigators and emergency personnel cooperate to investigate the chlorine leakage situation.
- A_3: Medical personnel come to the scene with medicine, e.g., *sodium bicarbonate solution and oxygen.*
- A_4: Emergency personnel deal with leaked chlorine.
- A_5: Slightly injured people are treated.
- A_6: Severely injured people are treated.
- A_7: Evaluate chlorine disposal.
- A_8: Perform post-treatment.
- r_1: Investigators.

TABLE I
ACTIVITY INFORMATION OF CHLORINE TANK EXPLOSION RESPONSE PROCESS

Activity	f_l	f_u	*Pre-Activity*	$f_{AR}=<r_1,r_2,r_3,r_4,r_5>$
A_1	10	12	Null	<8,4,0,0,0 >
A_2	3	7	Null	<12,8,0,0,5>
A_3	15	20	{A1}	<0,0,30,20,0>
A_4	10	15	{A2}	<0,0,0,0,0>
A_5	5	15	{A3}	<0,0,0,0,0>
A_6	10	20	{A3}	<0,0,0,0,0>
A_7	10	18	{A4}	<0,0,0,0,4>
A_8	1	2	{A5,A6}	<0,0,0,0,0>

TABLE II
RESOURCE INFORMATION OF CHLORINE TANK EXPLOSION RESPONSE PROCESS

Resource	f_l	f_u	*Quantity*	*Property*
*r*1	0.5	0.75	14	reusable
*r*2	1.25	1. 75	10	reusable
*r*3	1	1.2	30	consumable
*r*4	0.8	1.2	20	consumable
*r*5	1.4	2	5	reusable

r_2: Investigation equipment.
r_3: Sodium bicarbonate.
r_4: Medical oxygen.
r_5: Emergency personnel.

The activity information of this emergency response process is given in Table I, which contains duration constraints, relations among activities, and resource requirements. Resource information, i.e., the minimum and maximum preparation time, quantity and property, is shown in Table II. We have the following explanations for two Tables.

1) The process is composed of eight activities, denoted by $Activity = \{A_i | i \in Z_8\}$ where $Z_8 = \{1, 2, \ldots, 8\}$. They form certain relations. For example, from Table I, A_1 is a preactivity of A_3 (or A_3 is the post-activity of A_1), implying that A_3 cannot start before A_1.
2) The resources required are denoted by $Resource = \{r_j | j \in Z_5\}$. From Table I, both A_1 and A_2 require reusable resources r_1, i.e., r_1 are shared by A_1 and A_2. The total quantity of r_1 is less than the required one by A_1 and A_2, perhaps leading to a conflict if we plan to execute A_1 and A_2 simultaneously. As for consumable resource r_3 required by A_3, if its total quantity is less than that required by A_3, the whole process will pause.
3) For each activity, f_l represents the minimum execution duration, and f_u the maximum one. $f_l(A_1) = 10$ and $f_u(A_1) = 12$ mean that the minimum and maximum execution durations of A_1 are 10 and 12 time units, respectively. The timing function for each resource represents its minimum and maximum preparation durations.

C. E-Net for Emergency Response Process

In this section, we propose the formal definition of an E-Net. We assume that readers are familiar with the basic concepts of Petri nets [24]–[28], [35]. Some of the essential terminologies and notations are given as follows.

A tuple $N = (P, T; F)$ is named a net if the following conditions are satisfied: 1) $P \cap T = \varnothing$ and $P \cup T \neq \varnothing$; 2) $F \subseteq (P \times T) \cup (T \times P)$; and 3) $Dom(F) \cup Cod(F) = P \cup T$; where $Dom(F) = \{x \in P \cup T \mid \exists y \in P \cup T\colon (y, x) \in F\}$ and $Cod(F) = \{x \in P \cup T \mid \exists\, y \in P \cup T\colon (x, y) \in F\}$.

For all $x \in P \cup T$, the set ${}^{\cdot}x = \{y | y \in P \cup T \wedge (y, x) \in F\}$ is the preset of x, and $x^{\cdot} = \{y | y \in P \cup T \wedge (x, y) \in F\}$ is the post-set of x.

Definition 2 [32]: A *Petri net* is a 4-tuple $\Sigma = (P, T; F, M_0)$, where $N = (P, T; F)$ is a net, and $M_0\colon P \to Z$ is the initial marking of Σ.

Definition 3: $E = (P, T; F, M_0, \chi, \gamma, \alpha, \beta)$ is an *E-Net* if the following conditions are satisfied.

1) $(P, T; F, M_0)$ is a Petri net.
2) $P = P_A \cup \{p_s, p_e\}$ where P_A is an activity place set, and p_s and p_e are start and end places respectively.
3) $\chi\colon P_A \to Z^m$ is the resource function of activities. $\forall p \in P_A$, $\chi(p) = (q_1, q_2, \ldots q_m)$, where q_i is the quantity of r_i required by activity p and $r_i \in Resource$. *Resource* is the available resource set of an *E-Net*.
4) $\gamma\colon Resource \to \{0, 1\}$, $\gamma(r_i) = 0$ means that r_i is reusable, and $\gamma(r_i) = 1$ means that r_i is consumable.
5) $\alpha\colon P_A \cup Resource \to \mathbb{R}$. $\forall p \in P_A \cup Resource$, $\alpha(p) \geq 0$ is the minimum duration to execute/prepare activity/resource p.
6) $\beta\colon P_A \cup Resource \to \mathbb{R}$. $\forall p \in P_A \cup Resource$, $\beta(p) \geq 0$ is the maximum duration to execute/prepare activity/resource p, sastifying $\alpha(p) \leq \beta(p)$.
7) $\forall p \in P$, $M_0(p) = 1$ if ${}^{\cdot}p = \varnothing$, and otherwise $M_0(p) = 0$.

The firing rule of an E-Net is same as that of traditional Petri nets. Given a marking M, $\forall t \in T$, t is enabled under M if $\forall p \in {}^{\cdot}t$, $M(p) \geq 1$. Firing an enabed t removes a token from each of places in ${}^{\cdot}t$ and deposit one to each of places in $t^{\cdot}$. All properties, such as reachability and boundedness, can be defined similarly to those in a traditional Petri net. The main differences between E-Net and those for process modeling [15], [17]–[19] include the following.

1) To keep the atomic property of transitions, we use places to represent activities and their time is labeled on places to represent their execution duration.
2) Resource quantity is represented as a property function of an activity. According to their properties, resources are classified into reusable and consumable ones.
3) Each activity/resource has the minimum and maximum execution/preparation time. Timed Petri nets are proposed by introducing a firing time to each transition of a Petri net [23], [26], [47]. However, this transition firing duration totally violates the original atomic property of transitions in classical Petri nets [27]. Therefore, to keep the atomic property of transitions, we use places to represent activities and the time delays are associated with places to represent the activity execution duration. More studies that use places to stand for activities are conducted in [13], [14], [37], and [48]–[50]. To represent the uncertain activity execution duration, a timing interval including the minimum and maximum execution time, is introduced in our paper. Instead of using E-nets,

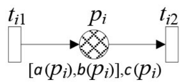


Fig. 1. E-Net for a single activity.

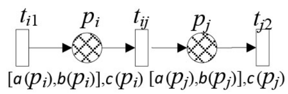


Fig. 2. E-Net for two sequential activities.

we may use stochastic Petri nets (SPN) [26], [46]. An SPN is a kind of timed Petri net models with delays obeying exponential distributions. However, it has the following disadvantages: 1) its analysis is via its conversion to and analysis of Markov chains, thereby easily leading to a state explosion problem and 2) the atomic property of transitions is violated. To avoid these problems, we use the time interval to stand for the duration of activities as often done by others studies [32], [49]–[61]. On the one hand, time interval allows for the uncertain execution duration of activities instead of exponentially or arbitrarily distributed random time delays. Other types of Petri nets [36], [38]–[40], [61], [62], such as predicate-transition nets, colored Petri nets, and labeled Petri nets, may be applied to characterize some more sophisticated modeling and analysis problems for emergency response processes. We defer their application to our future studies.

IV. Modeling Approaches

In a modeling procedure, we assume that sufficient resources are available to support the concurrent execution of activities requiring the same resources.

A. E-Net for Single Activity

A single activity is represented by one place and two transitions in an E-Net, as shown in Fig. 1. It is worth noting that activity places (with grids) are distinguished from logic ones (blank) in an E-net model according to their padding. Place p_i represents activity A_i, and transitions t_{i1} and t_{i2} represent the start and end of activity A_i. If p_i contains a token, its corresponding activity is on-going. Functions, such as $\chi(p_i)$, $\alpha(p_i)$, and $\beta(p_i)$ are labeled on p_i. In the following, p_i, also called the ith activity, means activity A_i.

B. E-Net for Emergency Response Process

An E-Net of a whole emergency response process can be obtained by the following constructs.

1) Assume that p_i is one of the preactivities of p_j, i.e., (p_i, p_j) $\subseteq$ *Relation*. Then t_{ij} is added between p_i and p_j to connect two sequential activities, as shown in Fig. 2. Transition t_{ij} represents both the end of activity p_i and the start of activity p_j.
2) Assume that p_i and p_j can execute concurrently. Then t_i and t_j are added to represent the start and end of p_i and p_j, as shown in Fig. 3.

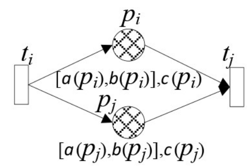


Fig. 3. E-Net for two parallel activities.

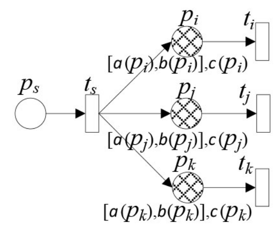


Fig. 4. E-Net for three activities without preactivities.

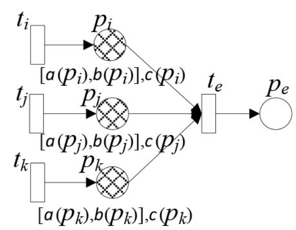


Fig. 5. E-Net for three activities without post-activities.

3) For activities without preactivities, we combine their start transitions to one, denoted as t_s and then add a start place p_s. They satisfy ${}^{\cdot}t_s = \{p_s\}$, $t_s{}^{\cdot} = \{p_i|\ p_i$ has no preactivities$\}$ and ${}^{\cdot}p_s = \{\varnothing\}$ and $p_s{}^{\cdot} = \{t_s\}$.

   An E-Net for three activities without preactivities is shown in Fig. 4 where p_s is a start place and t_s is a start transition of activities p_i, p_j, and p_k.
4) For activities without post-activities, we combine their end transitions to one, denoted as t_e and then add an end place p_e. They satisfy ${}^{\cdot}t_e = \{p_i|p_i$ has no post-activities$\}$, $t_e{}^{\cdot} = \{p_e\}$ and ${}^{\cdot}p_e = \{t_e\}$ and $p_e{}^{\cdot} = \{\varnothing\}$.

   An E-Net for three activities without post-activities is shown in Fig. 5 where p_e is an end place and t_e is the end transition of activities p_i, p_j, and p_k.
5) The initial marking M_0 of an E-Net satisfies: $M_0(p) = 1$ if $p = p_s$, and otherwise $M_0(p) = 0$.

   Algorithm 1 is given in the Appendix to transform an emergency response process specification into an E-Net automatically. Its complexity is mainly determined by its fourth step that takes $O(|P_A| \times (|P_A|-1)) \leq O(|P_A|^2)$. Thus, its complexity is $O(|P_A|^2)$ where $|P_A|$ is the number of activities.

Take the chlorine tank explosion response process described in Tables I and II as an example. By executing Algorithm 1, the E-Net is shown in Fig. 6.

V. Resource Configuration and Conflict Detection

In this section, resource configuration analysis and conflict detection for an emergency response process are performed.

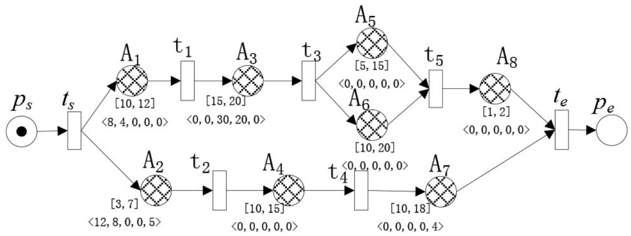


Fig. 6. E-Net of a chlorine tank explosion response process.

TABLE III
$T_{e1}(p)$, $T_{e2}(p)$, $T_{l1}(p)$, AND $T_{l2}(p)$ OF FIG. 6

activity	A_1	A_2	A_3	A_4	A_5	A_6	A_7	A_8	p_e
$T_{e1}(p)$	5	7	16	10	31	31	20	41	42
$T_{e2}(p)$	7	10	21	17	41	41	32	61	63
$T_{l1}(p)$	6	19	16	22	36	31	32	41	42
$T_{l2}(p)$	9	27	21	30	46	41	45	61	63

A. Ideal Execution Duration

Without considering resource conflicts or assuming no resource constraints, we can analyze the ideal execution duration of an emergency response process based on its E-Net.

If each activity is finished in its minimum duration, the earliest time to start activity p, denoted by $T_{e1}(p)$, is as follows:

$$T_{e1}(p) = \begin{cases} 0 & p = p_s \\ \max\{\{T_{e1}(p') + \alpha(p')|p' \in^{\bullet} ({}^{\bullet}p)\}, D_1(p)\} & \text{otherwise} \end{cases}$$

where $D_1(p) = max\{q(r_i) \times \alpha(r_i)|r_i \in \chi(p)\}$. D_1 represents the minimum preparation duration for resources required by p. Here, we assume that each kind of resources can be prepared simultaneously when the process starts.

If each activity is finished in its maximum duration, the earliest time to start activity p, denoted by $T_{e2}(p)$, is as follows:

$$T_{e2}(p) = \begin{cases} 0 & p = p_s \\ \max\{\{T_{e2}(p') + \alpha(p')|p' \in^{\bullet} ({}^{\bullet}p)\}, D_2(p)\} & \text{otherwise} \end{cases}$$

where $D_2(p) = max\{q(r_i) \times \beta(r_i)|r_i \in \chi(p)\}$. D_2 represents the maximum preparation duration for resources required by p.

Let $T_{E1} = T_{e1}(p_e)$ and $T_{E2} = T_{e2}(p_e)$, where p_e is the end place. T_{E1} and T_{E2} are the shortest execution durations when all activities are finished in their minimum and maximum durations, respectively. Moreover, T_{E1} is defined as the ideal execution time of the whole process if sufficient resources are provided, no resources conflict occurs, and all activities end in their minimum execution durations. In order to ensure the process to be finished in T_{E1}, the latest time to start activity p, denoted by $T_{l1}(p)$, is as follows:

$$T_{l1}(p) = \begin{cases} T_{E1}(p) & p = p_e \\ \min\{T_{l1}(p') - \alpha(p)|p' \in (p^{\bullet})^{\bullet}\} & \text{otherwise.} \end{cases}$$

In order to ensure the process to be finished in T_{E2}, the latest time to start activity p, denoted by $T_{l2}(p)$, is as follows:

$$T_{l2}(p) = \begin{cases} T_{E2}(p) & p = p_e \\ \min\{T_{l2}(p') - \beta(p)|p' \in (p^{\bullet})^{\bullet}\} & \text{otherwise.} \end{cases}$$

$T_{e1}(p)$, $T_{e2}(p)$, $T_{l1}(p)$, and $T_{l2}(p)$ for each activity in Fig. 6 are shown in Table III.

From Table III, we can obtain that the ideal execution duration of the chlorine tank explosion response process is 42 time units.

B. Resource Conflict Detection

The ideal execution duration of an emergency response process can be found. However, resource conflicts may exist during the process execution. Therefore, their detection methods are needed.

In the following discussion, let $E = (P, T; F, M_0, \chi, \gamma, \alpha, \beta)$ be an E-Net, and $T_{\text{start}}(p)$ and $T_{\text{end}}(p)$ represent the real start and end time of activity p. We have two types of resources, reusable and consumable. The former one becomes available and can be reused after being released while the latter is consumed during execution and cannot be reused. For this reason, resource conflict mentioned in our paper is essentially caused by reusable resources. If a conflict is caused by consumable resources, the only way to solve it is to add more such resources. For convenience, we first give the definition of reusable and consumable resource functions.

Definition 4: Given $p \in P_A$, define reusable resource vector $\Re(p)$ whose ith component is q_i if $\gamma(r_i) = 0$ and 0 otherwise and consumable resource vector $\varphi(p)$ whose ith component is q_i if $\gamma(r_i) = 1$ and 0 otherwise.

Clearly, resource function $\chi(p) = \Re(p) + \varphi(p)$.

Definition 5: $\forall p_i, p_j \in P_A$ $(p_i \neq p_j)$, p_i and p_j have resource dependency, denoted as $p_i \Theta p_j$, if $\Re(p_i)^T \cdot \Re(p_j) \neq \mathbf{0}$.

Definition 6: $\forall p_i, p_j \in P_A (p_i \neq p_j)$, p_i and p_j are in a potential resource conflict, denoted as $p_i \otimes p_j$, if 1) $p_i \Theta p_j$; and 2) $[T_{\text{start}}(p_i), T_{\text{end}}(p_i)]$ and $[T_{\text{start}}(p_j), T_{\text{end}}(p_j)]$ are overlapping.

Definition 6 defines a potential resource conflict, and this kind of conflicts can be avoided by providing sufficient resources. According to Definitions 5 and 6, we present Algorithm 2 to detect them in an E-Net.

In Algorithm 2, the complexity of Step 2 is $O(|P_A|^2)$. Because $O(|ConflictSet|) \leq O(|P_A|)$ where $|ConflictSet|$ represents the number of conficting activities, that of Step 3 is $O(|P_A|)$. Hence, its complexity is $O(|P_A|^2)$. It can be used to compute the potential conflict activity set, denoted as *ConflictSet*. However, even if there is $(p_i, p_j) \in ConflictSet$, we cannot say p_i and p_j are in a real conflict. It also depends on the total quantity of available resources committed to an emergency response process. Take the prior chlorine tank explosion response process as an example. We can see that A_1 and A_2 are in potential conflicts based on Algorithm 2. A_1 requires eight investigators and A_2 requires 12 investigators, and the total number of available investigators in this process is 14. Thus, conflict occurs because of the competition for investigators. If the total number of available investigators is more than 20, A_1 and A_2 will no longer be in conflict. If the total number of available investigators is less than eight, the emergency response process cannot be executed due to the shortage of investigators.

According to the aforementioned analysis, we can see that whether an emergency response process can be accomplished in its ideal execution duration depends on the total quantity of available resources. Next, we analyze the relationship between

TABLE IV
EXECUTION CASES OF AN EMERGENCY RESPONSE PROCESS

Case	Conditions	Effect
#1	$X_{AC}<X_{MC}$ or $X_{AR}<X_{MR}$	This is the worst case. An emergency response process cannot be performed due to lack of enough resources.
#2	$X_{AC}\geq X_{MC}$, $X_{AR}\geq X_{MR}$ and $X_{AR}<X_{RR}$	An emergency response process can be finished but the execution time may be delayed due to resource conflicts.
#3	$X_{AC}\geq X_{MC}$ and $X_{AR}\geq X_{RR}$	This is the best case. An emergency response process can be finished with its ideal execution duration.

the available resource quantity and execution of an emergency response process.

C. Resource Configuration Analysis

The available consumable and reusable resource vectors are denoted as X_{AC} and X_{AR}. We redefine several vector operators as follows. Let $X = (x_1, x_2, \ldots x_n)$ and $Y = (y_1, y_2, \ldots x_n)$ be two n-dimension vectors. If $\forall x_i \geq y_i$, for $i \in Z_n$, we denote $X \geq Y$ or $Y \leq X$. If $X \leq Y$ and $\exists i \in Z_n$, such that $x_i < y_i$, we denote $X < Y$.

The minimum consumable resource vector for an E-Net, named $X_{MC} = (q_1, q_2, \ldots q_k)$, can be obtained via Algorithm 3. Its complexity is $O(k \times |P_A|)$, where k is the number of consumable resource types.

X_{MC} has its real meaning, if $X_{AC} < X_{MC}$, the whole process will not be terminated because of shortage of consumable resources. The minimum reusable resource vector for an E-Net, denoted as $X_{MR} = (q_1, q_2, \ldots q_i)$, is obtained by Algorithm 4. Its complexity is $O(i \times |P_A|)$, where i is the number of reusable resource types.

If $X_{AR} < X_{MR}$, the whole process will not be terminated because of shortage of reusable resources. As mentioned in the last sub-section, even though $X_{AR} \geq X_{MR}$, resource conflicts may exist. This kind of conflicts may delay the execution duration of the whole process. Therefore, the reliable reusable resource vector X_{RR} is introduced. If $X_{AR} \geq X_{RR}$, there is no resource conflict during the process execution, i.e., potential resource conflicts disappear because sufficient resources are provided to support all the parallel activities.

The reliable reusable resource vector $X_{RR} = (q_1, q_2, \ldots q_j)$ is obtained via Algorithm 5. Its complexity is $O(j \times |P_A|)$ where j is the number of reusable resource types.

Based on the above analysis, we have three cases, i.e., the worst, delayed, and best one as shown in Table IV.

Take the chlorine tank explosion response process as an example. We can obtain $X_{MC} = <30, 20>$, $X_{MR} = <12, 8, 5>$, and $X_{RR} = <20, 12, 5>$ by executing Algorithms 3–5. From Table II, we have $X_{AC} = <30, 20>$ and $X_{AR} = <14, 10, 5>$. It is easy to conclude that it belongs to Case 2, i.e., it can be finished but the execution duration may be delayed due to resource conflicts. In this case, we must perform additional planning and analysis of the process execution.

TABLE V
$E_1(p)$, $E_2(p)$, $L_1(p)$, AND $L_2(p)$ OF FIG. 6

activity	A_1	A_2	A_3	A_4	A_5	A_6	A_7	A_8	p_e
$Wait_1(p,p_i)$	0	8	0	0	0	0	0	0	0
$E_1(p)$	5	15	16	18	31	31	28	41	42
$Wait_2(p,p_i)$	0	9	0	0	0	0	0	0	0
$E_2(p)$	7	19	21	26	41	41	41	61	63
$L_1(p)$	6	19	16	22	36	31	32	41	42
$L_2(p)$	9	27	21	30	46	41	45	61	63

VI. REAL EXECUTION DURATION ANALYSIS, OPTIMIZATION, AND PERFORMANCE EVALUATION

A. Earliest and Latest Time to Start Activity

In an emergency response process, an activity starts only after the termination of all its preactivities and under the condition that its required resources are available.

During the execution, even if each activity can be finished in its minimum duration, $T_{e1}(p)$ may not be the earliest time to start activity p because some activities prior to p may be delayed due to the wait for the required resources. The actual earliest time to start activity p is

$$E_1(p) = \begin{cases} 0 & p = p_s \\ T_{e1}(p) + W_1(p, p_1)|p \otimes p_1 & \text{otherwise} \end{cases}$$

where $W_1(p, p_1)$ is the waiting time of p for resources occupied by p_1 when p and p_1 are executed in their minimum duration

$$W_1(p, p_1) = \begin{cases} 0 & T_{e1}(p_1) + \alpha(p_1) \leq T_{e1}(p) \\ T_{e1}(p_1) + \alpha(p_1) - T_{e1}(p) & \text{otherwise} \end{cases}$$

where $\alpha(p)$ is the minimum execution duration of p. It is obvious that $E_1(p_e)$ is the shortest time to finish the process.

Similarly, if each activity is finished in its maximum duration, the earliest time to start activity p is

$$E_2(p) = \begin{cases} 0 & p = p_s \\ T_{e2}(p) + W_2(p, p_1)|p \otimes p_1 & \text{otherwise} \end{cases}$$

where $W_2(p, p_1)$ is the waiting time of p for resources occupied by p_1 when p and p_1 are executed in their maximum duration

$$W_2(p, p_1) = \begin{cases} 0 & T_{e2}(p_1) + \beta(p_1) \leq T_{e2}(p) \\ T_{e2}(p_1) + \beta(p_1) - T_{e2}(p) & \text{otherwise} \end{cases}$$

where $\beta(p)$ is the maximum execution duration of p.

$\forall p \in P_A$, $E_1(p)$ and $E_2(p)$ are the real earliest time to start p if all activities p_i before p can be finished in $\alpha(p_i)$ and $\beta(p_i)$. According to the aforementioned definitions of $E_1(p)$ and $E_2(p)$, it is obvious to conclude that $E_1(p) \leq E_2(p)$.

In the abovementioned timing analysis, we have only considered conflicts between two activities. Similarly, this conclusion can be easily extended to conflicts among three or more activities.

Denote $TE_1 = E_1(p_e)$ and $TE_2 = E_2(p_e)$ where p_e is the end place. Then TE_1 and TE_2 are the earliest time to finish the process if each activity is finished in its minimum and maximum durations, respectively. Also, there must be a latest time to start each activity to ensure the execution of the whole process to be finished in TE_1 or TE_2.

TABLE VI
COMPARISON OF REAL START TIME BETWEEN TWO RESOLUTION METHODS

	activity	A_1	A_2	A_3	A_4	A_5	A_6	A_7	A_8	p_e
Mehod 1	$Wait_1(p,p_i)$	0	8	0	0	0	0	0	0	0
	$E_1(p)$	5	15	16	18	31	31	28	41	42
Mehod 2	$Wait_1(p,p_i)$	5	0	0	0	0	0	0	0	0
	$E_1(p)$	10	7	21	10	36	36	20	46	47

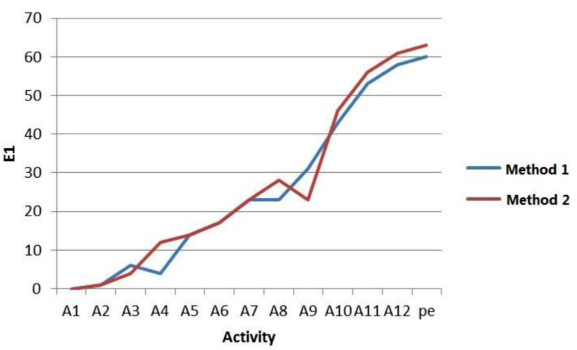


Fig. 7. Performance comparison of resolution methods.

TABLE VII
ACTIVITY INFORMATION

Activity	f_l	f_u	Pre-Activities	f_{AR}=<r_1,r_2,r_3,r_4>
A_1	1	2	Null	<0,0,0,0>
A_2	3	7	{A_1}	<0,0,0,0>
A_3	8	10	{A_2}	<8,12,0,0>
A_4	2	4	{A_2}	<10,16,0,0>
A_5	3	8	{A_3, A_4}	<0,0,0,0>
A_6	6	8	{A_5}	<0,0,0,0>
A_7	20	28	{A_6}	<0,0,30,0>
A_8	18	20	{A_6}	<0,0,0,6>
A_9	5	6	{A_6}	<0,0,0,10>
A_{10}	10	20	{A_7, A_8, A_9}	<0,0,20,0>
A_{11}	5	8	{A_{10}}	<0,0,0,0>
A_{12}	2	3	{A_{11}}	<0,0,0,0>

TABLE VIII
RESOURCE INFORMATION

Resource	f_l	f_u	Quantity	Property
r_1	0.2	0.25	12	reusable
r_2	0.1	0.15	18	reusable
r_3	0.5	0.75	50	consumable
r_4	0.3	0.35	14	reusable

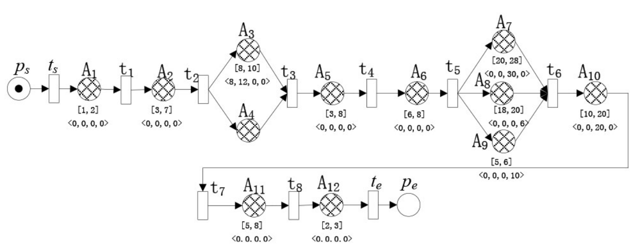


Fig. 8. E-Net of chlorine explosion case.

To ensure an emergency response process to be finished in TE_1, we have the latest time to start activity p

$$L_1(p) = \begin{cases} E_1(p) & p = p_e \\ \min\{L_1(p_1) - \alpha(p) | p' \in (p^\bullet)^\bullet\} & \text{otherwise.} \end{cases}$$

To ensure an emergency response process to be finished in TE_2, we have the latest time to start activity p

$$L_2(p) = \begin{cases} E_2(p) & p = p_e \\ \min\{L_2(p') - \beta(p) | p' \in (p^\bullet)^\bullet\} & \text{otherwise.} \end{cases}$$

B. Priority-Activity-First Strategy

A resolution strategy is needed to remove resource conflicts and achieve the highest performance. Traditional resource conflict resolution strategies, such as key-activity priority, waiting-short priority, and start-early priority ones, are used to remove resource conflicts [14]. However, a different resolution strategy is related to a different execution duration. To finish an emergency response process as early as possible, we propose a priority-activity-first strategy.

Definition 7: Given $p_i \,\Theta\, p_j$, if $p_i \in PriorityActivitySet$, but $p_j \notin PriorityActivitySet$, priority p_i is higher than p_j, i.e., $W(p_i, p_j) = 0$ and $W(p_j, p_i) > 0$, where $W(p_i, p_j)$ is the waiting time of p_i for p_j. This is called a priority-activity-first strategy.

Same as a key-activity priority strategy (KAPS), the proposed one also defines two priority levels for all activities. The former one has key level and non-key level while the proposed one has priority and non-priority levels. Different levels are determined according to different algorithms. The algorithm to obtain key activities is discussed in [14]. For the proposed one, an activity with high priority is supposed to execute first when it is in conflict with others. Activities with high priority can be obtained through Algorithm 6.

We have the following remarks about Algorithm 6. BufferRange(p_i) represents the maximum buffer time of p_i. BufferRange(p_i) > BufferRange(p_j) means that p_i can hold longer periods than p_j to keep the whole process duration invariant. If p_j runs first, then $W(p_i, p_j)$ − BufferRange(p_i) means that the real delay of the whole process is caused by p_i waiting for p_j. If $W(p_i, p_j)$ − BufferRange(p_i) ≥ $W(p_j, p_i)$ − BufferRange(p_j), then the delay of p_i will cause longer delay of the whole process than that of p_j. Thus, we set p_i with higher priority than p_j, i.e., p_i is executed first to lead to the better overall performance. The complexity of Algorithm 6 is $O(|P_A|^2)$.

Based on Algorithm 6, we can obtain the priority activities from activities in conflicts. Take the chlorine tank explosion process as an example. PriorityActivitySet = {A_1} is obtained by Algorithm 6. Based on the priority-activity-first strategy, the optimized $E_1(p)$, $E_2(p)$, $L_1(p)$, and $L_2(p)$ for the E-Net in Fig. 6 is calculated and shown in Table V.

TABLE IX

$T_{e1}(p)$, $T_{e2}(p)$, $T_{l1}(p)$, AND $T_{l2}(p)$ OF FIG. 8

Activity	A_1	A_2	A_3	A_4	A_5	A_6	A_7	A_8	A_9	A_{10}	A_{11}	A_{12}	p_e
$T_{e1}(p)$	0	1	4	4	12	15	21	21	21	41	51	56	58
$T_{e2}(p)$	0	2	9	9	19	27	35	35	35	63	63	71	74
$T_{l1}(p)$	0	1	4	10	12	15	21	23	26	41	51	56	58
$T_{l2}(p)$	0	2	9	15	19	27	35	29	37	43	63	71	74

TABLE X

COMPARISON OF START TIME BETWEEN METHODS 1 AND 2

	activity	A_1	A_2	A_3	A_4	A_5	A_6	A_7	A_8	A_9	A_{10}	A_{11}	A_{12}	p_e
Mehod 1	$Wait_1(p,p_i)$	0	0	2	0	0	0	0	0	8	0	0	0	0
	$E_1(p)$	0	1	6	4	14	17	23	23	31	43	53	58	60
Mehod 2	$Wait_1(p,p_i)$	0	0	0	8	0	0	0	5	0	0	0	0	0
	$E_1(p)$	0	1	4	12	14	17	23	28	23	46	56	61	63

Compared with Table III, we can conclude that the real execution duration of the emergency response process is same as that of the ideal one for this particular case, i.e., the priority-activity-first strategy can effectively resolve resource conflicts and optimize the process performance.

C. Evaluation of Priority-Activity-First Strategy

Next, we compare the priority-activity-first strategy called Method 1 with an existing one. According to Zeng *et al.* [14], the method by combining waiting-short priority strategy (WSPS) and KAPS called Method 2 can ensure the process to finish in a shorter time than WSPS and start-early priority strategy used alone. Thus, we compare the proposed one with Method 2 only. For convenience, we use the minimum execution duration for each activity in the following analysis.

We first define the WSPS and KAPS following [14].

Definition 8: WSPS: $\forall p_i \Theta p_j$, if $W(p_i, p_j) \le W(p_j, p_i)$, let $W(p_i, p_j) = 0$. Otherwise, let $W(p_j, p_i) = 0$.

Definition 9: KAPS: $\forall p_i \Theta p_j$, if p_i is a key activity but p_j is not, the priority of p_i is higher than p_j, i.e., $W(p_i, p_j) = 0$ and $W(p_j, p_i) > 0$.

According to Zeng *et al.* [14], we can obtain the key activity set of the process in Fig. 6 from Table III, KeyActivity = $\{p_s, A_3, A_6, A_8, p_e\}$.

Then, a comparison between Methods 1 and 2 in terms of performance is obtained. The real start time of each activity is illustrated in Table VI and Fig. 7.

Based on the comparison, we can conclude that our priority-activity-first strategy can guarantee the whole process to finish in a shorter time than Method 2 for this process.

VII. CASE STUDY

To illustrate the modeling and analysis approach for an emergency response process based on E-Nets, we have presented a simplified chlorine tank explosion response process in Section III. Its detailed one is described as follows: the related departments report the chlorine leakage. The investigators investigate the injuries and chlorine leakage. Medical personnel come to the scene with medical equipment and medicine. Emergency personnel deal with leaked chlorine. Some of the involved activities and resources in this example are described as follows.

- A_1: Publish the chlorine leakage news.
- A_2: Rescue team rush to the site.
- A_3: Investigators investigate the injuries.
- A_4: Investigators investigate the chlorine leakage situation.
- A_5: Summary the investigation results.
- A_6: Make effective disposal plan.
- A_7: Emergency personnel deal with leaked chlorine.
- A_8: Treatment to slightly injured people.
- A_9: Treatment to severely injured people.
- A_{10}: Conducting hazard mitigation operations.
- A_{11}: Emergency evaluation.
- A_{12}: Process post-treatment.
- r_1: Investigators.
- r_2: Investigation equipment.
- r_3: Slaked lime.
- r_4: Doctors and nurses.

Detailed information about activities and resources is given in Tables VII and VIII.

Through Algorithm 1, we convert an emergency response process to an E-Net automatically, as shown in Fig. 8. Then, we give the ideal earliest and latest start time for each activity as shown in Table IX. From Table IX, we can obtain the ideal execution duration for the process is 48 time units. Via Algorithm 2, we identify that A_3 and A_4 are in conflicts and so are A_8 and A_9. By executing Algorithms 3–5, we obtain $X_{MC} = <50>$, $X_{MR} = <10, 16, 10>$, and $X_{RR} = <18, 28, 16>$. From Table IX, we have $X_{AC} = <50>$ and $X_{AR} = <12, 18, 14>$. According to Table V, it is easy to conclude that this process is a delayed case.

Next, we conduct the real time analysis of this process. By Algorithm 6, we can obtain the priority activities from conflicting activities, as PriorityActivitySet = $\{A_4, A_8\}$. Then, Methods 1 and 2 are compared. The real start time of each activity is illustrated in Table X and Fig. 9.

Based on the comparison between Methods 1 and 2, it is easy to see that our strategy can ensure an emergency response process to finish in a shorter time than

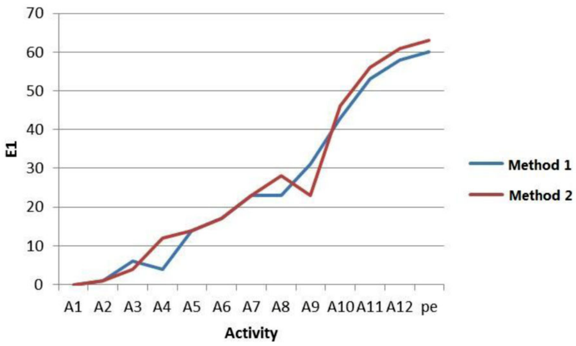


Fig. 9. Performance comparison between Methods 1 and 2.

Method 2. Hence, the priority-activity-first strategy should be adopted.

VIII. Conclusion

An emergency response process is a special process that differs from a traditional business one [13]–[22]. There are limited formal methods to model and analyze it. This paper presents an E-Net to do so. Its resource and duration management with the help of its E-Net model can be performed. The main contributions of this paper include:

1) The formal specification for an emergency response process constrained by resources and uncertain durations is introduced.
2) An E-Net based on Petri nets is proposed to model this kind of emergency response processes.
3) Ideal execution duration of an emergency response process is derived based on its E-Net model.
4) Conflict detection algorithms are proposed to detect potential resource conflicts.
5) According to resource availability, the execution of an emergency response process is classified into the worst, delayed, and best cases.
6) For a delayed case, the earliest and latest time to start each activity is found. A priority-activity-first strategy is proposed and can achieve a better execution result than the state-of-the-art methods.

This work only gives a kind of static analysis, i.e., our analysis is conducted before the start of an emergency response process, which is based on the specification at build-time and uses fixed data. Our future work should focus on dynamic analysis, i.e., the actual execution time (in a time interval) can be obtained and used instead of those pregiven ones. Then, we can analyze how executed activities influence the execution of the remaining activities dynamically. This paper also opens the door to the following future research.

1) An activity executer has not been fully considered. A formal model based on predicate-transition nets [36], colored Petri nets [38], and labeled Petri nets [39], [40] may be needed to handle some sophisticated processes.
2) A complex emergency response management system is often executed by geographically dispersed partners or different organizations. As a solution for dealing with its decentralized nature, a complicated process can be fragmented into small pieces and scheduled to different servers for its execution [41]. Such fragmentation algorithms should be developed for it.
3) Deadlock exists during the execution of the complex emergency response processes with shared resources. Advanced deadlock control methods [33], [42], [43], [63]–[67] should be introduced for an E-Net.
4) An emergency response process may contain choice and other complicated structures. Modeling and analysis methods for such processes are highly desired.
5) Arbitrary distributions are used for time delay [26], [46]. Their use in E-Nets to model stochastic emergency response processes should be pursued.

Appendix

E-Net Transformation Algorithm

Algorithm 1: Transform an Emergency Response Process to an E-Net

INPUT: $ERP = <$ *Activity, Resource, Time, Rproperty, Relation*, f_{AR}, f_{RP}, f_l, $f_u >$.

OUTPUT: $E = (P, T; F, M_0, \chi, \gamma, \alpha, \beta)$.

/*Step 1: initialization*/

Step 1: $P \leftarrow \varnothing$, $T \leftarrow \varnothing$, $F \leftarrow \varnothing$, $M_0 \leftarrow 0$, $\chi \leftarrow \varnothing$, $\alpha \leftarrow \varnothing$ and $\beta \leftarrow \varnothing$;

/*Step 2: to assign activity set to P_A*/

Step 2: $P_A \leftarrow Activity$ and $P \leftarrow P \cup P_A$;

/*Step 3: to add time and resource functions to place in P_A*/

Step 3: For i to $|P_A|$, DO

(3.1) $\chi(p_i) \leftarrow f_{AR}(p_i)$;

(3.2) $\alpha(p_i) \leftarrow f_l(p_i)$;

(3.3) $\beta(p_i) \leftarrow f_u(p_i)$;

END DO

/*Step 4: to add transition to connect sequential activities*/

Step 4: For two places $p_i, p_j \in P_A$ with precedence relation, DO

(4.1) $T \leftarrow T \cup \{t_{ij}\}$;

(4.2) $F \leftarrow F \cup \{(p_i, t_{ij}), (t_{ij}, p_j)\}$;

END DO

/*Step 5: to add transition to connect activity followed by multiple activities*/

Step 5: For a place p followed by multiple parallel ones, such as $p_i, p_j \ldots p_k \in P$, DO

(5.1) $T \leftarrow T \cup \{t\}$;

(5.2) $F \leftarrow F \cup \{(p, t), (t, p_i), (t, p_j) \ldots (t, p_k)\}$;

END DO

/*Step 6: to add transition to connect activity which follows multiple parallel activities*/

Step 6: For a place p which follows multiple parallel ones, such as $p_i, p_j \ldots p_k \in P$, DO

(6.1) $T \leftarrow T \cup \{t\}$;

(6.2) $F \leftarrow F \cup \{(t, p), (p_i, t), (p_j, t) \ldots (p_k, t)\}$;

END DO

/*Step 7: to add start place and start transition for activities without preactivities*/

Step 7: For places without preactivities, such as $p_i, p_j \dots p_k \in P$

(7.1) $T \leftarrow T \cup \{t_s\}$;

(7.2) $P \leftarrow P \cup \{p_s\}$;

(7.3) $\chi(p_s) \leftarrow \mathbf{0}$, $\alpha(p_s) \leftarrow 0$, $\beta(p_s) \leftarrow 0$;

(7.4) $F \leftarrow F \cup \{(p_s, t_s), (t_s, p_i), (t_s, p_j) \dots (t_s, p_k)\}$;

/*Step 8: to add end place and end transition for activities without post-activities*/

Step 8: For places without post-activities, such as $p_i, p_j \dots p_k \in P$

(8.1) $T \leftarrow T \cup \{t_e\}$;

(8.2) $P \leftarrow P \cup \{p_e\}$;

(8.3) $\chi(p_e) \leftarrow \mathbf{0}$, $\alpha(p_e) \leftarrow 0$, $\beta(p_e) \leftarrow 0$;

(8.4) $F \leftarrow F \cup \{(t_e, p_e), (p_i, t_e), (p_j, t_e) \dots (p_k, t_e)\}$;

/*Step 9: to add initial marking to start place p_s*/

Step 9: $M_0(p_s) = 1$;

Step 10: Output $E \leftarrow (P, T; F, M_0, \chi, \gamma, \alpha, \beta)$.

Resource Conflict Detection Algorithm

Algorithm 2: Detect Potential Resource Conflicts in an E-Net

INPUT: $E = (P, T; F, M_0, \chi, \gamma, \alpha, \beta)$.

OUTPUT: *ConflictSet* $= \{(p_i, p_j) | p_i \otimes p_j\}$

/*Step 1: initialization*/

Step 1: *ConflictSet* $\leftarrow \varnothing$, $T_{e1}(p_s) \leftarrow 0$, $T_{e2}(p_s) \leftarrow 0$, $T_{l1}(p_s) \leftarrow 0$, and $T_{l2}(p_s) \leftarrow 0$;

/*Step 2: to detect resource dependency among activities*/

Step 2: FOR $\forall p_i, p_j \in P_A (p_i \neq p_j)$ DO

IF $\Re(p_i)^T \cdot \Re(p_j) \neq \mathbf{0}$ THEN

ConflictSet $\leftarrow$ *ConflictSet* $\cup \{(p_i, p_j)\}$;

END IF

END DO

/*Step 3: to detect if time interval of activities with resource dependency are overlapping*/

Step 3: IF *ConflictSet* $\neq \varnothing$ THEN

For $\forall (p_i, p_j) \in$ *ConflictSet* DO

(3.1) Calculate $T_{e1}(p_i)$, $T_{e2}(p_i)$, $T_{e1}(p_j)$, and $T_{e2}(p_j)$;

(3.2) IF $([T_{e1}(p_i), T_{e2}(p_i) + \beta(p_i)] \cap [T_{e1}(p_j), T_{e2}(p_j) + \beta(p_j)] \neq \varnothing)$ THEN

GOTO Step 3;

ELSE *ConflictSet* $\leftarrow$ *ConflictSet* $- \{(p_i, p_j)\}$;

END IF

END DO

END IF

Step 4: Output *ConflictSet*

Resource Configuration Algorithm

Algorithm 3: Calculate the Minimum Consumable Resource Vector X_{MC}

INPUT: $E = (P, T; F, M_0, \chi, \gamma, \alpha, \beta)$.

OUTPUT: X_{MC}.

/*Step 1: initialization*/

Step 1: $sum \leftarrow 0$, $X_{MC} \leftarrow \mathbf{0}$;

/*Step 2: to obtain the minimum consumable resource vector*/

Step 2: FOR $\forall X_{MC}.q_i \in X_{MC}$ DO

(2.1) FOR $\forall p_i \in P_A$ DO

$sum \leftarrow sum + \varphi(p_i).q_i$;

END DO

(2.2) $X_{MC}.q_i \leftarrow sum$;

(2.3) $sum \leftarrow 0$;

END DO

Step 3: Output X_{MC}.

Algorithm 4: Calculate the Minimum Reusable Resource Vector X_{MR}

INPUT: $E = (P, T; F, M_0, \chi, \gamma, \alpha, \beta)$.

OUTPUT: X_{MR}.

/*Step 1: initialization*/

Step 1: $X_{MR} \leftarrow \mathbf{0}$;

/*Step 2: to obtain the minimum reusable resource vector*/

Step 2: FOR $\forall X_{MR}.q_i \in X_{MR}$ DO

FOR $\forall p_i \in P_A$ DO

IF $\Re(p_i).q_i > X_{MR}.q_i$ THEN

$X_{MR}.q_i \leftarrow \Re(p_i).q_i$

END IF

END DO

END DO

Step 3: Output X_{MR}.

Algorithm 5: Calculate the Reliable Reusable Resource Vector X_{RR}

INPUT: $E = (P, T; F, M_0, \chi, \gamma, \alpha, \beta)$, *ConflictSet* and X_{MR}.

OUTPUT: X_{RR}.

/*Step 1: initialization*/

Step 1: $sum \leftarrow 0$, $X_{RR} \leftarrow X_{MR}$;

/*Step 2: to obtain the reliable reusable resource vector*/

Step 2: FOR $\forall X_{RR}.q_i \in X_{RR}$ DO

FOR $\forall (p_i, p_j) \in$ *ConflictSet* DO

(2.1) $sum \leftarrow \Re(p_i).q_i + \Re(p_j).q_i$;

(2.2) IF $sum > X_{MR}.q_i$ THEN

$X_{RR}.q_i \leftarrow sum$;

END IF

(2.3) $sum \leftarrow 0$;

END DO

END DO

Step 3: Output X_{RR}.

Priority Activity Obtain Algorithm

Algorithm 6: Obtain Priority Activities

INPUT: *ConflictSet*, $E = (P, T; F, M_0, f_{AR}, l, \alpha, \beta)$

OUTPUT: *PriorityActivitySet*

/*Step 1: initialization*/

Step 1: *PriorityActivitySet* $\leftarrow \varnothing$, *BufferRange*$(p_s) \leftarrow 0$, $W(p_i, p_j) \leftarrow 0$, $T_{e1}(p_s) \leftarrow 0$, $T_{l1}(p_s) \leftarrow 0$;

/*Step 2: to obtain the *PriorityActivitySet**/

Step 2: FOR $\forall (p_i, p_j) \in$ *ConflictSet* DO

(2.1) Calculate $T_{e1}(p_i)$, $T_{l1}(p_i)$, $T_{e1}(p_j)$, $T_{l1}(p_j)$, $W(p_i, p_j)$, and $W(p_j, p_i)$;

(2.2) *BufferRange*$(p_i) \leftarrow T_{l1}(p_i) - T_{e1}(p_i)$;

BufferRange$(p_j) \leftarrow T_{l1}(p_j) - T_{e1}(p_j)$;

(2.3) IF $W(p_i, p_j) -$ *BufferRange*$(p_i) \geq$

$W(p_j, p_i) -$ *BufferRange*(p_j) THEN

PriorityActivitySet $\leftarrow$ *PriorityActivitySet* $\cup \{p_i\}$;

ELSE

PriorityActivitySet $\leftarrow$ *PriorityActivitySet* $\cup \{p_j\}$;

END IF
END DO
Step 3: Output *PriorityActivitySet*.

REFERENCES

[1] K. Kang, "Application of code approach for emergency evacuation in a rail station," *Fire Technol.*, vol. 43, no. 4, pp. 331–346, 2007.

[2] J. Wang, "Emergency healthcare workflow modeling and timeliness analysis," *IEEE Trans. Syst., Man, Cybern. A, Syst., Humans*, vol. 42, no. 6, pp. 1323–1331, Nov. 2012.

[3] G. Chu and J. Wang, "A stochastic analysis on occupant evacuation time under emergency conditions," in *Proc. 2011 2nd IEEE Int. Conf. Emergency Manag. Manag. Sci.*, Beijing, China.

[4] L. Chen and Q. Tang, "Path selection algorithms for emergency logistics based time expectation," in *Proc. Int. Conf. Logist. Eng. Manag.*, Chengdu, China, 2010.

[5] J. Li and C. Li, "Simulation and analysis of time-consumption of emergency supplies loading services based on queue theory," in *Proc. Multimodal Approach Sustained Transp. Syst. Develop. Inf., Technol., Implement.*, Wuhan, China, 2011.

[6] H. Cheng and X. Yang, "Theory model for earthquake emergency shelter choice based on the shortest overall evacuation time," in *Proc. Multimodal Approach Sustained Transp. Syst. Develop. Inf., Technol., Implement.*, Wuhan, China, 2011.

[7] F. Fiedrich, F. Gehbauer, and U. Rickers, "Optimized resource allocation for emergency response after earthquake disasters," *Safety Sci.*, vol. 35, no. 1, pp. 41–57, 2000.

[8] J. Tsenga, M. Liub, R. Changc, J. Sud, and C. Shua, "Emergency response plan of chlorine gas for process plants in Taiwan," *J. Loss Prev. Process Ind.*, vol. 21, no. 4, pp. 393–399, 2008.

[9] J. Wang, W. Tepfenhart, and D. Rosca, "Emergency response workflow resource requirements modeling and analysis," *IEEE Trans. Syst., Man, Cybern. C, Appl. Rev.*, vol. 39, no. 3, pp. 270–283, May 2009.

[10] C. Sell and I. Braun, "Using a workflow management system to manage emergency plans," in *Proc. 6th Int. ISCRAM Conf.*, Gothenburg, Sweden, May 2009.

[11] A. Chen and F. Peña-Mora, "Decentralized approach considering spatial attributes for equipment utilization in civil engineering disaster response," *J. Comput. Civil Eng.*, vol. 25, no. 6, pp. 457–470, 2011.

[12] M. Reznek *et al.*, "Emergency medicine crisis resource management (EMCRM): Pilot study of a simulation-based crisis management course for emergency medicine," *Acad. Emerg. Med.*, vol. 10, no. 2, pp. 386–389, 2003.

[13] H. Wang and Q. Zeng, "Modeling and analysis for workflow constrained by resources and nondetermined time: An approach based on Petri nets," *IEEE Trans. Syst., Man, Cybern. A, Syst., Humans*, vol. 38, no. 4, pp. 802–817, Jul. 2008.

[14] Q. Zeng, H. Wang, D. Xu, H. Duan, and Y. Han, "Conflict detection and resolution for workflows constrained by resources and non-determined durations," *J. Syst. Softw.*, vol. 819, pp. 1491–1504, Sep. 2008.

[15] J. Li, Y. Fan, and M. Zhou, "Performance modeling and analysis of workflow," *IEEE Trans. Syst., Man, Cybern. A, Syst., Humans*, vol. 34, no. 2, pp. 229–242, Mar. 2004.

[16] G. Janssens, J. Verelst, and B. Weyn, "Techniques for modeling workflows and their support for reuse," in *Business and Process Management: Models, Techniques and Empirical Studies,* W. van der Aalst, Ed. Berlin, Germany: Springer, 2000, pp. 1–15.

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

[18] N. R. Adam, V. Atluri, and W. K. Huang, "Modeling and analysis of workflows using Petri nets," *J. Intell. Inf. Syst.*, vol. 10, no. 2, pp. 131–158, 1998.

[19] J. Li, Y. Fan, and M. C. Zhou, "Timing constraint workflow nets for workflow analysis," *IEEE Trans. Syst., Man, Cybern. A, Syst., Humans*, vol. 33, no. 2, pp. 179–193, Mar. 2003.

[20] J. Li, Y. Fan, and M. C. Zhou, "Approximate performance analysis of workflow model," in *Proc. IEEE Int. Conf. Syst., Man Cybern.*, vol. 2. Oct. 2003, pp. 1175–1180.

[21] S. Ling and H. Schmidt, "Time Petri nets for workflow modelling and analysis," in *Proc. IEEE Int. Conf. Syst., Man Cybern.,* Nashville, TN, USA, 2000, pp. 3039–3044.

[22] W. Tan and M. C. Zhou, *Business and Scientific Workflows: A Service-Oriented Approach*. Hoboken, NJ, USA: Wiley, 2013.

[23] M. C. Zhou and K. Venkatesh, *Modeling, Simulation and Control of Flexible Manufacturing Systems: A Petri Net Approach*. Singapore: World Scientific, 1998.

[24] R. Zurawski and M. C. Zhou, "Petri Nets and Industrial Applications: A Tutorial," *IEEE Trans. Ind. Electron.*, vol. 41, no. 6, pp. 567–583, Dec. 1994.

[25] W. Reisig, *Understanding Petri Nets: Modeling Techniques, Analysis Methods, Case Studies*. Berlin, Germany: Springer, 2013.

[26] J. Wang, *Timed Petri Nets: Theory and Application*. Boston, MA, USA: Kluwer Academic Publishers, 1998.

[27] C. A. Petri, "Kommunikation mit Automaten," Ph.D. dissertation Instrum. Math., Univ. Darmstadt, Bonn, Germany, 1962.

[28] W. Reisig, *Petri Nets: An Introduction*. Berlin, Germany: Springer-Verlag, 1985.

[29] Y. Y. Du, C. J. Jiang, and M. C. Zhou, "A petri nets based correctness analysis of internet stock trading systems," *IEEE Trans. Syst., Man, Cybern. C, Appl. Rev.*, vol. 38, no. 1, pp. 93–99, Jan. 2008.

[30] Y. Y. Du, C. J. Jiang, and M. C. Zhou, "A petri net-based model for verification of obligations and accountability in cooperative systems," *IEEE Trans. Syst., Man, Cybern. A, Syst., Humans*, vol. 39, no. 2, pp. 299–308, Mar. 2009.

[31] B. Hruz and M. C. Zhou, *Modeling and Control of Discrete Event Dynamic Systems*. London, U.K., Springer, 2007.

[32] N. Q. Wu and M. C. Zhou, *System Modeling and Control with Resource-Oriented Petri Nets*. New York, NY, USA: CRC Press, 2010.

[33] M. C. Zhou and M. P. Fanti, *Deadlock Resolution in Computer-Integrated Systems*. New York, NY, USA: Marcel Dekker, Jan. 2005.

[34] W. M. P. Aalst, "Three good reasons for using a Petri-net-based workflow management system," in *Proc. Int. Work. Conf. Inf. Process Integr. Enterprises*, Cambridge, MA, USA, Nov. 1996, pp. 179–201.

[35] C. Liu, Q. Zeng, J. Zou, F. Lu, and Q. Wu, "Invariant decomposition conditions for Petri nets based on the index of transitions," *Inf. Technol. J.*, vol. 11, no. 7, pp. 768–774, 2012.

[36] T. Murata and D. Zhang, "A predicate-transition net model for parallel interpretation of logic programs," *IEEE Trans. Softw. Eng.*, vol. 14, no. 4, pp. 481–497, Apr. 1988.

[37] S. X. Sun, Q. Zeng, and H. Wang, "Process-mining-based workflow model fragmentation for distributed execution," *IEEE Trans. Syst., Man, Cybern. A, Syst., Humans*, vol. 41, no. 2, pp. 294–310, Mar. 2011.

[38] R. Robidoux, H. Xu, L. Xing, and M. C. Zhou, "Automated verification of dynamic reliability block diagrams using colored Petri nets," *IEEE Trans. Syst., Man, Cybern. A, Syst., Humans*, vol. 40, no. 2, pp. 337–351, Mar. 2010.

[39] Y. Y. Du, L. Qi, and M. C. Zhou, "A vector matching method for analyzing logic Petri nets," *Enterprise Inf. Syst.*, vol. 5, no. 4, pp. 449–468, 2011.

[40] Y. Y. Du, L. Qi, and M. C. Zhou, "Analysis and application of logical Petri nets to e-Commerce systems," *IEEE Trans. Syst., Man, Cybern., Syst.*, vol. 44, no. 4, pp. 468–481, Apr. 2014.

[41] W. Tan and Y. Fan, "Dynamic workflow model fragmentation for distributed execution," *Comput. Ind.*, vol. 58, no. 5, pp. 381–391, 2007.

[42] Z. W. Li and M. C. Zhou, *Deadlock Resolution in Automated Manufacturing Systems: A Novel Petri Net Approach*. New York, NY, USA: Springer, 2009.

[43] Z. W. Li, N. Q. Wu, and M. C. Zhou, "Deadlock control of automated manufacturing systems based on Petri nets—A literature review," *IEEE Trans. Syst., Man, Cybern. C, Appl. Rev*, vol. 42, no. 4, pp. 437–462, Jul. 2012.

[44] W. Kim and M. Chung, "A web service support to collaborative process with semantic information," in *Proc. 6th Int. Conf. Web Inf. Syst. Eng.*, Berlin, Germany: Springer-Verlag, 2005, pp. 217–230.

[45] W. M. P. Aalst, "Interval timed colored Petri nets and their analysis," Berlin, Germany: Springer, 1993.

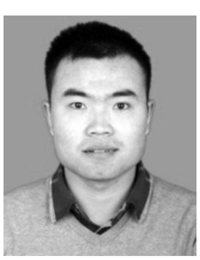
[46] F. Bause and P. S. Kritzinger, *Stochastic Petri Nets—An Introduction to the Theory* (Advanced Studies in Computer Science). Wiesbaden, Germany: Verelag Vieweg, 1996.

[47] Y.-S. Huang, Y.-S. Weng, and M. C. Zhou, "Modular design of urban traffic-light control systems based on synchronized timed Petri nets," *IEEE Trans. Intell. Transp. Syst.*, vol. 5, no. 2, pp. 530–539, Apr. 2014.

[48] Q. Zeng, F. Lu, C. Liu, and D. Meng, "Modeling and analysis for cross-organizational emergency response systems using Petri nets," *Chinese J. Comput.*, vol. 36, no. 11, pp. 2290–2302, 2013.

[49] H. H. Xiong and M. C. Zhou, "Scheduling of semiconductor test facility via Petri nets and hybrid heuristic search," *IEEE Trans. Semicond. Manuf.*, vol. 11, no. 3, pp. 384–393, Aug. 1998.

[50] H. Hu, M. C. Zhou, and Z. Li, "Algebraic synthesis of timed supervisor for automated manufacturing systems using Petri nets," *IEEE Trans. Autom. Sci. Eng.*, vol. 7, no. 3, pp. 549–557, Jul. 2010.
[51] Y. Qiao, N. Wu, and M. C. Zhou, "Scheduling of dual-arm cluster tools with wafer revisiting and residency time constraints," *IEEE Trans. Ind. Informat.*, vol. 10, no. 1, pp. 286–300, Feb. 2014.
[52] N. Wu, M. C. Zhou, F. Chu, and C. Chu, "A Petri-net-based scheduling strategy for dual-arm cluster tools with wafer revisiting," *IEEE Trans. Syst., Man, Cybern. Syst.*, vol. 43, no. 5, pp. 1182–1193, Sep. 2013.
[53] N. Wu, M. C. Zhou, and F. Chu, "A Petri net based heuristic algorithm for realizability of target refining schedules in oil refinery," *IEEE Trans. Autom. Sci. Eng.*, vol. 5, no. 4, pp. 661–676, Oct. 2008.
[54] N. Wu and M. C. Zhou, "A closed-form solution for schedulability and optimal scheduling of dual-arm cluster tools with wafer residency time constraint based on steady schedule analysis," *IEEE Trans. Autom. Sci. Eng.*, vol. 7, no. 2, pp. 303–315, Apr. 2010.
[55] N. Wu, F. Chu, C. Chu, and M. C. Zhou, "Petri net-based scheduling of single-arm cluster tools with reentrant atomic layer deposition processes," *IEEE Trans. Autom. Sci. Eng.*, vol. 8, no. 1, pp. 42–55, Jan. 2011.
[56] N. Wu and M. C. Zhou, "Schedulability analysis and optimal scheduling of dual-arm cluster tools with residency time constraint and activity time variation," *IEEE Trans. Autom. Sci. Eng.*, vol. 9, no. 1, pp. 203–209, Jan. 2012.
[57] L. Pan, Z. Ding, and M. C. Zhou, "A configurable state class method for temporal analysis of time Petri nets," *IEEE Trans. Syst., Man, Cybern. Syst.*, vol. 44, no. 4, pp. 482–493, Apr. 2014.
[58] Y. Du, W. Tan, and M. C. Zhou, "Timed compatibility analysis of web service composition: A modular approach based on Petri nets," *IEEE Trans. Autom. Sci. Eng.*, vol. 11, no. 2, pp. 594–606, Apr. 2014.
[59] Q. Zhu, N. Wu, Y. Qiao, and M. C. Zhou, "Petri net-based optimal one-wafer scheduling of single-arm multi-cluster tools in semiconductor manufacturing," *IEEE Trans. Semicond. Manuf.*, vol. 26, no. 4, pp. 578–591, Dec. 2013.
[60] W. Tan, Y. Fan, and M. C. Zhou, "A Petri net-based method for compatibility analysis and composition of web services in business process execution language," *IEEE Trans. Autom. Sci. Eng.*, vol. 6, no. 1, pp. 94–106, Jan. 2009.
[61] N. Wu and M. C. Zhou, "Colored timed Petri nets for modeling and analysis of cluster tools," *Asian J. Control*, vol. 12, no. 3, pp. 253–266, May 2010.
[62] W. Tan, Y. Fan, M. C. Zhou, and Z. Tian, "Data-driven service composition in building SOA solutions: A Petri net approach," *IEEE Trans. Autom. Sci. Eng.*, vol. 7, no. 3, pp. 686–694, Jul. 2010.
[63] J. Li, M. C. Zhou, X. Dai, and Y. Gan, "Robust control reconfiguration of resource allocation systems with Petri nets and integer programming," *Automatica*, vol. 50, no. 1, pp. 915–923, Jan. 2014.
[64] H. Liu, K. Xing, M. Zhou, L. Han, and F. Wang, "Transition cover-based design of Petri net controllers for automated manufacturing systems," *IEEE Trans. Syst., Man, Cybern. A, Syst. Humans*, vol. 44, no. 2, pp. 196–208, Feb. 2014.
[65] Y. Chen, Z. Li, and M. C. Zhou, "Optimal supervisory control of flexible manufacturing systems by Petri nets: A set classification approach," *IEEE Trans. Autom. Sci. Eng.*, vol. 11, no. 2, pp. 549–563, Apr. 2014.
[66] M. P. Fanti and M. C. Zhou, "Deadlock control methods in automated manufacturing systems," *IEEE Trans. Syst., Man, Cybern. A, Syst. Humans*, vol. 34, no. 1, pp. 5–22, Jan. 2004.
[67] Z. W. Li, G. Y. Liu, M-H. Hanisch, and M. C. Zhou, "Deadlock prevention based on structure reuse of Petri net supervisors for flexible manufacturing systems," *IEEE Trans. Syst., Man, Cybern. A, Syst. Humans*, vol. 42, no. 1, pp. 178–191, Jan. 2012.



Cong Liu received the B.S. degree in computer science and technology from Shandong University of Science and Technology, Qingdao, China, in 2013, and is currently pursuing the master's degree from the College of Information Science and Engineering, Shandong University of Science and Technology.

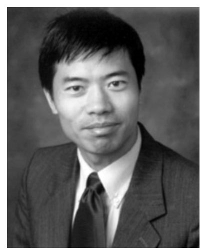
His current research interests include Petri nets, service composition, and business process management.

Qingtian Zeng received the Ph.D. degree in computer software and theory from the Institute of Computing Technology, Chinese Academy of Sciences, Beijing, China, in 2005.

He was a Visiting Professor at the City University of Hong Kong, Kowloon, Hong Kong, in 2008. He is currently a Professor with Shandong University of Science and Technology, Qingdao, China. His current research interests include Petri nets, process mining, and knowledge management.

Hua Duan received the Ph.D. degree in applied mathematics from Shanghai Jiaotong University, Shanghai, China, in 2008.

She is an Associate Professor at Shandong University of Science and Technology, Qingdao, China. Her current research interests include Petri nets, process mining, and machine learning.



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

He joined the New Jersey Institute of Technology, Newark, NJ, USA, in 1990, and is now a Distinguished Professor of Electrical and Computer Engineering. He is currently a Professor with Tongji University, Shanghai, China. His current research interests include Petri nets, sensor networks, web services, semiconductor manufacturing, transportation, and energy systems. He has over 500 publications including 11 books, 250+ journal papers (majority in IEEE Transactions), and 22 book-chapters.

Prof. Zhou is the Founding Editor of IEEE Press Book Series on Systems Science and Engineering, and an Associate Editor of the IEEE TRANSACTIONS ON SYSTEMS, MAN, AND CYBERNETICS: SYSTEMS, IEEE TRANSACTIONS ON INDUSTRIAL INFORMATICS, and IEEE TRANSACTIONS ON INTELLIGENT TRANSPORTATION SYSTEMS. He is a Life Member of the Chinese Association for Science and Technology-USA and served as its President in 1999. He is a Fellow of IFAC and AAAS.

Faming Lu received the Ph.D. degree in computer software and theory from Shandong University of Science and Technology, Qingdao, China, in 2012.

He is currently a Lecturer with Shandong University of Science and Technology. His current research interests include Petri nets, process modeling, and process mining.

Jiujun Cheng received the Ph.D. degree from the Beijing University of Posts and Telecommunications, Beijing, China, in 2006.

He is currently an Associate Professor at Tongji University, Shanghai, China. He received the support from Research Fund for the Doctoral Program of Higher Education and Young Excellent Talents in Tongji University. His current research interests include P2P overlay and social networks with a focus on mobile/Internet interworking, service composition, and Internet of vehicles.