Formulating the Data-Flow Modeling and Verification for Workflow: A Petri Net based Approach

Cong LIU
Department of Computer Science, Shandong University of Science and Technology, Qingdao 266590, China

Qingtian ZENG*
Department of Computer Science, Shandong University of Science and Technology, Qingdao 266590, China

Hua DUAN
Department of Mathematics, Shandong University of Science and Technology, Qingdao 266590, China

Abstract: Control-flow and data-flow modeling and verification are two important challenges for traditional workflow management. To formulate the data-flow modeling and verification, a Petri Net based approach is proposed in this paper. More specifically, the WFIO-net, which is a kind of WF-net by extending each activity with its read and write data sets, is first presented to model both the control-flow and data-flow information in a workflow process. Moreover, the firing rule of WFIO-net which is different from traditional Petri net is discussed in detail. Then the formal definitions of three basic types of data-flow errors, including missing data, redundant data and conflicting data are formulated using WFIO-net. Next, a polynomial complexity algorithm which is capable of detecting the above-mentioned three kinds of basic data-flow anomalies is proposed based on the activity-data incidence matrix of the WFIO-net. A running case of a property loan approval business process is given to validate the proposed method.

Keywords: workflow data-flow modeling and verification; Petri net, WFIO-net and its firing rule, activity-data incidence matrix

1. INTRODUCTION

Workflow Management, which focuses on the automation of business processes, has received considerable attention in recent years due to its potential for significantly increasing productivity and saving costs. A workflow is a representation of a given process that is made up of well-defined collection of activities, also referred to as tasks. In recent days, workflow structural correctness verification (also refers to as control-flow perspective) [1]-[2] has received much more attention than the other aspects. However, a complete workflow description requires to characteristic the other perspectives, including time, resource, data and etc. Time management [3]-[4] is an important aspect for workflow modeling, analysis and verification. In addition to structural and temporal constraints, resource constraints are also of vital importance in workflow management, because activities in a workflow usually need to access some resources during their execution [5]-[6]. Similarly, the role of data is also important, e.g. routing choices in a process are typically determined by certain data element, which indicates that the control-flow, to some extent, is data-dependent. Unfortunately, the data-flow can be erroneous itself. Towards this issue, data-flow modeling and correctness verification are highly intensified in [7]-[12].

As a tool to model and analyze physical systems, Petri nets [13]-[16] have shown great power in dealing with concurrences and conflicts. As mentioned by Aalst [1] there are at least three reasons for using Petri nets to model and analyze workflows: (1) Graphical nature and formal semantics have been defined for Petri nets; (2) Petri nets are state-based instead of event-based, so the state of the case can be modeled explicitly in Petri nets; and (3) Petri nets are characterized by the availability of many analysis techniques. In this paper, we formulate the data-flow modeling and verification for workflows based on a kind of extended Petri net with data elements. More specifically, our research goal is to develop a systemic formal methodology for characterizing the data-flow modeling and verification. The contribution of this paper is two-fold. In the first place, Petri net-based model for workflow in terms of control-flow and data-flow are presented to formulate the workflow execution semantic. Secondly, formal specifications of three kinds of data-flow errors and corresponding detection algorithm are given, based on which their correctness verification is conducted.

The remainder of this paper is organized as follows. Section 2 discusses the related work. Section 3 introduces a typical scenario of a property loan approval business process. In Section 4, the WFIO-net, which is a WF-net by extending each activity with its read and write data sets, is proposed to model the data-flow aspect. Section 5 addresses the verification of three kinds of data-flow errors using the activity-data incidence matrix of the WFIO-net. Finally, Section 6 draws conclusions and points our future research scope.

2. RELATED WORK

The importance of data-flow verification in workflow is first mentioned in [7]. There several potential data-flow problems are identified. However, no formal models as well as modeling approaches are presented and only basic data errors are defined, however, no formal validation methods are involved. Another idea is to integrate data-flow into process models. The idea behind this approach is to define an extension for a workflow model so that data-flow can be shown with the control-flow. Based on Petri Net formalism containing three building blocks, storage, reactive and transformational units, an extended version of Dual Workflow Nets is proposed in [8]. Unfortunately, this model is complex even for small running-examples and no indication about how this model can be created. Du et al. proposed an approach called Data-Flow Skeleton Filled with Activities (DFSFA) in [9]. Basically, the workflow process is derived from the data-flow skeleton and then is filled with activities. More specifically, a data-flow dependency tree is first generated based on the data dependency. Next step is to generate the data-flow skeleton and then fills it with activities. A more complete approach is provided in [10], where a data-flow matrix and various relations between data and activities are defined. Then, some data-flow errors are conceptualized using UML diagrams and supporting verification algorithms are discussed. This work was further extended and generalized in

[11]. More recently, a survey on data-flow modeling of business processes is presented by DOLEAN and PETRUSEL [12]. They pointed that (1) data is essential in a process and activities need data; (2) without data or without data available on time, the control-flow cannot be executed; (3) there is no model showing a clear data-flow perspective focusing on how data changes during a process executing.

According to these literature reviews, it is easy to conclude that existing researches on data-flow modeling and verification suffer at least the following limitations: (1) no formal model with executing semantic is addressed. Although the model in [8] is proposed by extending Petri net, the firing rule is destroyed; and (2) no formal verification is investigated based on formal models.

3. A RUNNING EXAMPLE

A typical scenario of a property loan approval business process mainly includes the following steps: (1) To start the application, an application is received; (2) Then, the completeness of the application is verified; (3) To determine the applicant's qualifications, the financial service company first verifies the applicant's employment status; (4) To qualify the applicant, the financial service company checks the applicant's credit history; (5) At the same time, the financial company also checks the applicant's liquid assets; (6) If the applicant is qualified, the interest rate is determined in for a certain period; (7) The loan application is then evaluated; (8) If the applicant agrees with everything, the applicant signs the application; (9) The loan officer for signature.

According to the abovementioned steps, we can distil the activity information, including their name, meaning, connection relation and read/write data set is shown in Table 1. Table 2 gives the meaning of each data elements.

4. MODELING APPROACHES BASED WFIO-NET

Modeling approaches of business processes have been studied for decades, and some excellent models such as, WF-net, XPDL, BPMN and etc. [18], have been widely applied. Our work is based on Petri net, WF-net to be more precise. We assumed that readers are familiar with the basic concepts of WF-net [1]-[2] and Petri nets [13]-[18]. Some of the essential terminologies and notations are listed 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) $F\subseteq(P\times T)\cup(T\times P)$; (3) $Dom(F)\cup Cod(F)=P\cup T$; where $Dom(F)= \{x\in P\cup T |\exists y\in P\cup T\text{: } (y,x)\in F\}$ and $Cod(F)= \{x\in P\cup T|\exists\ y\in P\cup T\text{: } (x, y)\in F\}$. For all $x\in P\cup T$, the set ${}^{\bullet}x=\{y|\ y\in P\cup T\wedge(y, x)\in F\}$ is the pre-set of x, and $x^{\bullet}=\{y|\ y\in P\cup T \wedge(x, y) \in F\}$ is the post-set of x.

Definition 4.1 A *Petri net* is a 4-tuple $\Sigma=(P, T; F, M_0)$, where

Table 1 Activity Information of this Property Loan Approval Process

Activity Name	Meaning	Pre-activities	Write Date	Read Data
A_1	Receive application	$\varnothing$	$\{D_1, D_2, D_3\}$	$\varnothing$
A_2	Verify completeness of the application	$\{A_1\}$	$\{D_4\}$	$\{D_1, D_2, D_3\}$
$A3$	Verify employment status	$\{A_2\}$	$\varnothing$	$\{D_5\}$
$A4$	Check credit history	$\{A_2\}$	$\varnothing$	$\{D_5\}$
A_5	Verify liquid asset	$\{A_2\}$	$\varnothing$	$\{D_5\}$
A_6	Determine interest rate	$\{A_3, A_4, A_5\}$	$\{D_5\}$	$\{D_6\}$
A_7	Evaluate loan application	$\{A_6\ \}$	$\{D_6, D_7\}$	$\{D_8\}$
A_8	Contact applicant for agreement	$\{A_7\}$	$\{D_8\}$	$\{D_9\}$
A_9	Forward to loan officer for signature	$\{A_8\}$	$\{D_9\}$	$\{D_{10}\}$

Table 2 Data Information of this Property Loan Approval Process

Data Name	Meaning	Data Name	Meaning
D_1	Applicant name	D_6	Interest rate
D_2	Loan amount	D_7	Current owner of property
D_3	Annual income	D_8	Evaluation result
D_4	Application complete	D_9	Signed by applicant
D_5	Applicant qualified	D_{10}	Signed by loan officer

$N=(P, T; F)$ is a net, and M_0: $P\to Z^+$ is the initial marking of Σ where $M(p)$ represents the number of tokens in place p and Z^+ is a non-negative integer set.

We usually use a rectangle to represent a transition, a circle to represent a place, and a dot to represent a token. An initial marking is denoted by M_0. p is marked by M iff $M(p)>0$. A transition $t\in T$ is enabled under M, if and only if $\forall p\in {}^\bullet t$: $M(p)>0$, denoted as $M[t>$. If $M[t>$ holds, t may fire, resulting in a new marking M', denoted as $M[t>M'$, such that $M'(p)=M(p)-1$ if $\forall p\in {}^\bullet t \setminus t^\bullet$, $M'(p)=M(p)+1$ if $\forall p\in t^\bullet \setminus {}^\bullet t$, and otherwise $M'(p)=M(p)$.

A Petri net which models a workflow process is called the workflow net (WF-net) whose definition is summarized following [1]-[2].

Definition 4.2 A *Petri net* $\Sigma=(P, T; F, M_0)$ is a WF-net if: (1) There is one source place $p_s\in P$ such that ${}^\bullet p_s=\varnothing$; (2) There is one sink place $\in P$ such that $p_e^\bullet=\varnothing$; (3) Each node $x\in P\cup T$ is on a path from p_s to p_e; and (4) $\forall p\in P$, $M_0(p)=1$ if $p=p_s$, and otherwise $M_0(p)=0$.

The WF-net is capable to model the control of business where the transition set T is used to represent the activities, and source place and sink place represent the start and end of the process respectively. The WF-net model of the property loan approval business process in Section 3 is shown in Figure 1. This workflow is composed of nine activities, denoted by $t_{Ai}=\{i|i\in Z_9\}$ where $Z_9=\{1,2,\ldots,9\}$.

However, the WF-net can only represent the logic relation of workflow, i.e. the control-flow aspect. To model the data-flow elements of a workflow, we propose the WFIO-net, which is a WF-net by extending each activity with its input and output data sets.

Definition 4.3 $\Sigma_{IO}=(P, T; F, I, O, M_0)$ is a WFIO-net if the following conditions are satisfied:
(1) $P=P_D\cup P_L$ and $P_D\cap P_L=\varnothing$ where P_D is a data place set and P_L is a logic place set;
(2) $T=T_A\cup T_L$ and $T_A\cap T_L=\varnothing$ where T_A is an activity transition set and T_L is a logic transition set;
(3) $F=F_L\cup F_D$ where $F_L\subseteq(P_L\times T_L)\cup(T_L\times P_L)$ is the logic flow and $F_D\subseteq(P_D\times T_D)\cup(T_D\times P_D)$ is the data flow;
(4) I: $T_A\to P_D$. $\forall t_a\in T_A$, $I(t_a)$ is the input data set of activity t_a; and
(5) O: $T_A\to P_D$. $\forall t_a\in P_A$, $O(t_a)$ is the output data set of activity t_a; and (6) $\forall p\in P$, $M_0(p)=1$ if $p=p_s$, otherwise $M_0(p)=0$.

Control-flow structure can be represented by the $(P_L, T_L; F_L, M_0|_L)$ which is same with the traditional WF-net. The main differences between WFIO-net and WF-net lie in: (1) Two components are involved in the place set, i.e., $P=P_L\cup P_D$, data place set (P_D) which is drawn in double circles is separated from the logic place set (P_L); and (2) The firing rule of the transition is different from that of a standard one. All data elements are initially undefined, thereby each data place does not contain any token in the model before execution.

There two kinds of flow relation in a WFIO-net, one kind to represent token flow (F_L) which is same with the classical one and the other kind is used to represent the data flow (F_D). Graphically, a token flow is drawn by direct arc end with arrow while the data flow is represented with direct arc end with small circle. The pre-set of a transition t contains two parts, formally denoted as ${}^\bullet t={}^\lozenge t\cup{}^\circ t$ where ${}^\lozenge t$ is the logic pre-set of t while ${}^\circ t$ is its data pre-set of t (or read place set). Similarly, the post-set of a transition t also contains two parts, formally denoted as $t^\bullet=t^\lozenge\cup t^\circ$ where $t^\lozenge$ is the logic post-set of t while t° is its data post-set (or write place set). For example, the Figure 2 (a) gives a simple transition model of WFIO-net, we have its pre-set is ${}^\bullet t_a=\{p_1, p_{read}\}$, logic pre-set ${}^\lozenge t_a=\{p_1\}$ and read place ${}^\circ t_a=\{p_{read}\}$. The post-set of t_a, denoted as $t_a^\bullet=\{p_2, p_{write}\}$, logic post-set $t_a^\lozenge=\{p_2\}$ and write place $t_a^\circ=\{p_{write}\}$.

A transition $t\in T$ in a WFIO-net is enabled under M, if and only if $\forall p\in {}^\bullet t$: $M(p)>0$, denoted as $M[t>$. If $M[t>$ holds, t may fire, resulting in a new marking M', denoted as $M[t>M'$, such that $M'(p)=M(p)-1$ if $\forall p\in {}^\lozenge t \setminus t^\lozenge$, $M'(p)=M(p)+1$ if $\forall p\in t^\bullet \setminus {}^\bullet t$, and otherwise $M'(p)=M(p)$.

A WFIO-net model for single activity is drawn in Figure 2 and the following explanations are given: (1) p_{read} and p_{write}

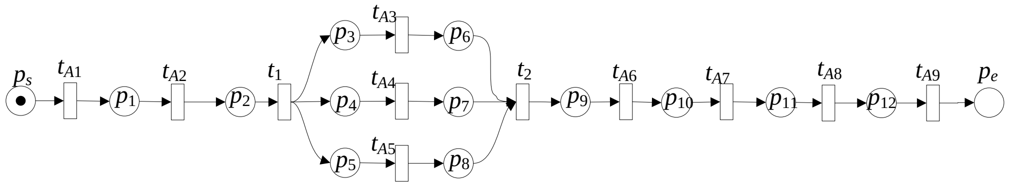

Figure 1 WF-net of the Property Loan Approval Business Process

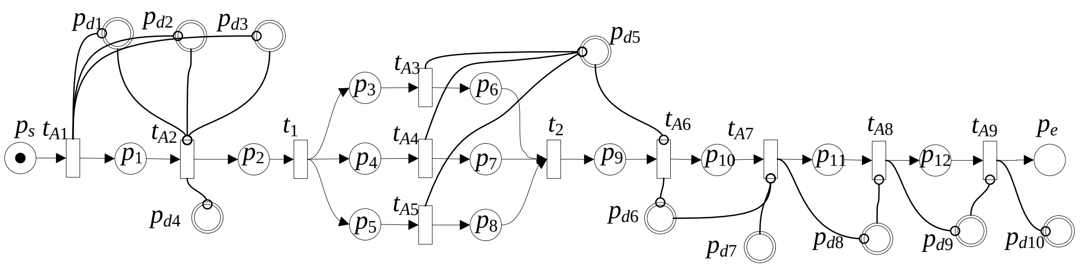

Figure 3 WFIO-net of the Property Loan Approval Business Process

are used to represent the data places and p_1 and p_2 are used to represent the logic places; (2) t_a is enabled in Figure 2 (a) as both p_1 and p_{read} has token; and (3) Figure 2 (b) shows the state after firing t_a where the token in p_1 is removed to p_2 and p_{write} obtains one token without removing the token in p_{ready}.

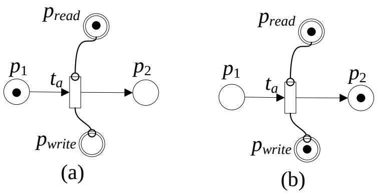

Figure 2 WFIO-net of Single Activity (a) Enabled State, (b) Fired State

The WFIO-net model of the property loan approval business process in Section 3 is demonstrated in Figure 3. The involved data elements are denoted by $p_{dj}=\{j|\ j\in Z_{10}\}$, where $Z_{10}=\{1,2,\ldots,10\}$.

5. VERIFICATION OF DATA-FLOW BASED ON WFIO-NET

In this section, data-flow verification of workflow is performed on the basis of WFIO-net. As proved by Sun *et al.* [10], missing data, redundant data and conflicting data are defined as the basic data-flow anomalies. Other types of data-flow errors that are proposed by Sadiq *et al.* [7] can be represented by these three basic ones. Therefore, we restrict our verification scope to these three basic data-flow errors.

(1) *Missing Data*
When a data element is accessed without initialization, in this case a missing data anomaly occurs, which is the same as the variable use without definition and initialization in programming. Its formal definition is given in Definition 5.1.

Definition 5.1 Let $\Sigma_{IO}=(P, T; F, I, O, M_0)$ be a WFIO-net, $\forall p_d\in P_D$, p_d is a missing data if (1) $^{\circ}p_d=\emptyset$; and (2) $p_d{}^{\circ}\neq\emptyset$.

In Definition 5.1, the first criterion means that the data element is not initialized by an activity, and the second criterion shows this data will be used by an activity. Missing data can cause system exception and dangling and therefore should be detected before system enactment.

(2) *Redundant Data*
If an activity produces data items that do not contribute to the production of the final output data, then there is a redundant data anomaly, i.e. a data element is produced by one activity but never used by other activities. This is same as the variable defined but never used in programming.

Definition 5.2 Let $\Sigma_{IO}=(P, T; F, I, O, M_0)$ be a WFIO-net, $\forall p_d\in P_D$, p_d is a redundant data if (1) $^{\circ}p_d\neq\emptyset$; and (2) $p_d{}^{\circ}=\emptyset$.

In Definition 5.2, the first criterion means that the data element is initialized by an activity, and the second criterion shows this data will never be used by an activity. Redundant data can cause system inefficiency and therefore should be detected before system enactment.

(3) *Conflicting Data*
In a workflow instance, if there exist different versions of the same data element, conflicting data anomalies occur. For example, more than one activity attempts to initialize (write) the same data element in one workflow instance.

Definition 5.3 Let $\Sigma_{IO}=(P, T; F, I, O, M_0)$ be a WFIO-net, $\forall p_d\in P_D$, p_d is a conflicting data if (1) $^{\circ}p_d\neq\emptyset$; (2) $p_d{}^{\circ}\neq\emptyset$; and (3) $|^{\circ}p_d|\geq 2$.

In Definition 5.3, the first and third criteria show that the data element is initialized by more than one activity, and the second criterion shows this data will be used by an activity. Conflicting data can cause uncertainty or even confusion and therefore should be detected before system enactment.
In the following, we propose an approach to detect these three basic kinds of data-flow errors. Before rendering our specific detection algorithm, we would like to introduce the activity-task incidence matrix of a WFIO-net.

According to [13], any Petri net can be represented as an incidence matrix. Traditional incidence matrix shows the control-flow relation but no data-flow informal is reflected.

Definition 5.4 Assume that M is an $m\times n$ (m rows, n columns) matrix, where m is the number of transitions (activities) and n is the number of data places in a WFIO-net. For each position $[i, j]$ in the matrix, place a "1" in the position if transition i has input to position j, and place a "-1" in the position if transition i has output from position j, and A "0" is placed in the position if position i does not have input from position j. In this way, M is defined as the *Activity-Data Incidence Matrix* of WFIO-net.

According to Definition 5.4, if "1" is placed at position $[i, j]$, it represents that the ith activity writes a data to the jth data element. Similarly, if "-1" is placed at position $[i, j]$, it means that the ith activity reads a data from the jth data element. The activity-data incidence matrix of the WFIO-net in Fig. 3 is constructed in the following.

$$\begin{array}{c|cccccccccc|}
 & p_{d1} & p_{d2} & p_{d3} & p_{d4} & p_{d5} & p_{d6} & p_{d7} & p_{d8} & p_{d9} & p_{d10} \\
t_{A1} & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
t_{A2} & -1 & -1 & -1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\
t_{A3} & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\
t_{A4} & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\
t_{A5} & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\
t_{A6} & 0 & 0 & 0 & 0 & -1 & 1 & 0 & 0 & 0 & 0 \\
t_{A7} & 0 & 0 & 0 & 0 & 0 & -1 & -1 & 1 & 0 & 0 \\
t_{A8} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 1 & 0 \\
t_{A9} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 1
\end{array}$$

Figure 4 Activity-Data Incidence Matrix of WFIO-net in Figure 3

Here we propose Algorithm 1 to detect dataflow errors, including missing data error, redundant data error and conflicting data error in a WFIO-net based on the activity-data incidence matrix of the WFIO-net.

Algorithm 1: Detect Data-flow Errors in a WFIO-net.
INPUT: The activity-data incidence matrix M of a WFIO-net.
OUTPUT: *MisDataSet*, *RedDataSet* and *ConDataSet*.
Step 1: *MisDataSet* ←∅, *RedDataSet* ←∅, and *ConDataSet*←∅, *ReadNum*←0 and *WriteNum*←0;

```
Step 2: FOR $j$=0 to $j$=$|P_d|$-1 DO
        (1) FOR $i$=0 to $|T_a|$-1 DO
                IF $p_{ij}$==1 THEN
                        WriteNum++;
                ELSE IF $p_{ij}$==-1 THEN
                        ReadNum++;
                END IF
                END DO
        (2) IF WriteNum==0 and ReadNum>0 THEN
                $MisDataSet \leftarrow MisDataSet \cup \{p_{dj}\}$;
        ELSE IF WriteNum>0 and ReadNum==0 THEN
                $RedDataSet \leftarrow RedDataSet \cup \{p_{dj}\}$;
        ELSE IF WriteNum>1 and ReadNum>0 THEN
                $ConDataSet \leftarrow ConfDataSet \cup \{p_{dj}\}$;
        END IF
        (3) ReadNum←0 and WriteNum←0;
        END DO
Step 3: Output MisDataSet, RedDataSet, and ConfDataSet.
```

The complexity of Algorithm 1 is mainly determined by the second step whose complexity is $O(|P_d|\times|T_a|)$. Thus, the complexity of Algorithm 1 is $O(|P_d|\times|T_a|)$, where $|P_d|$ is the number of data places and $|T_a|$ is the number of activities. Take the property loan approval business process in Section 3 as an example. By executing Algorithm 1, we can obtain *MisDataSet*=$\{p_{d7}\}$, *RedDataSet* =$\{p_{d4}, p_{d10}\}$ and *ConDataSet* =$\{p_{d5}\}$.

6. CONCLUSIONS

To formulate the data-flow modeling and verification for workflow management, a Petri Net based approach is proposed. The contributions of this work are summarized as follows: (1) to model the data-flow elements of a workflow, we propose the WFIO-net, which is a WF-net by extending each activity with its read and write data sets; and (2) The formal definitions of three basic types of data-flow errors, including missing data, redundant data and conflicting data are given based on WFIO-net and their corresponding detection algorithm is also addressed using the activity-data incidence matrix of WFIO-net. However, this is our first work towards formal modeling and verification of data-flow anomalies. More efforts will be highly desired at least in the following two aspects：(1) Detailed taxonomy of each kind of data-flow error will be discussed. For example, the missing data error may contain several sub-classes such absolute missing and conditional missing as mentioned in [10]; and (2) Time factor will be introduced to our WFIO-net to give a more accurate verification manner, i.e. write operation to one data element is not permitted when it is reading/writing by the other activity.

7. ACKNOWLEDGEMENTS

This work was supported in part by NSFC (61170079 and 61202152), by the Special Fund for Agro-scientific Research in the Public Interest (201303107), by the special Fund for Fast Sharing of Science Paper in Net Era by CSTD (2013122), by the Sci. & Tech. Development Fund of Qingdao (13-1-4-153-jch), by the open project of the Key Laboratory of Embedded System and Service Computing, Ministry of Education, Tongji University (ESSCKF201403), the Excellent Young Scientist Foundation of Shandong Province (BS2012DX030 and ZR2013FQ030) and the Graduate Innovation Foundation Project of Shandong University of Science and Technology (YC140106).

8. REFERENCES

[1] W. M. P. van der Aalst, “The application of Petri nets to workflow management,” *Journal of Circuits, Systems and Computers*, vol. 8, no. 1, pp. 21–66, 1998.

[2] W. M. P. van der Aalst, K. M. van Hee, A. H. M. ter Hofstede et al., “Soundness of workflow nets: classification, decidability, and analysis,” *Formal Aspects of Computing*, vol. 23, no. 3, pp. 333–363, 2011.

[3] Jianqiang Li, Yushun Fan, Mengchu Zhou, “Performance modeling and analysis of workflow”, IEEE Transactions on Systems, Man and Cybernetics, Part A: Systems and Humans, vol.34, no.2, pp. 229- 242, March 2004.

[4] Jianqiang Li, Yushun Fan, Mengchu Zhou, “Timing Constraint Workflow Nets for Workflow Analysis”, IEEE Transactions on Systems, Man, and Cybernetics-Part A: Systems and Humans, Vol.33, No.2, March 2003, pp.179-193.

[5] Huaiqing Wang, Qingtian Zeng, “Modeling and Analysis for Workflow Constrained by Resources and Nondetermined Time: An Approach Based on Petri Nets”, IEEE Transactions on Systems, Man and Cybernetics, Part A: Systems and Humans, vol.38, no.4, pp.802-817, July 2008.

[6] Qingtian Zeng, Huaiqing Wang, Dongming Xu, Hua Duan, Yanbo Han, “Conflict detection and resolution for workflows constrained by resources and non-determined durations”, Journal of Systems and Software, 819: 1491-1504, 2008.

[7] Shazia Sadiq, Maria Orlowska, Wasim Sadiq, and Cameron Foulger. 2004. Data flow and validation in workflow modelling. In *Proceedings of the 15th Australasian database conference - Volume 27* (ADC '04), Klaus-Dieter Schewe and Hugh Williams (Eds.), Vol. 27. Australian Computer Society, Inc., Darlinghurst, Australia, Australia, 207-214.

[8] Shaokun Fan, Wanchun Dou, Jinjun Chen. Dual Workflow Nets: Mixed Control/Data-Flow Representation for Workflow Modeling and Verification, Advances in Web and Network Technologies, and Information Management, Lecture Notes in Computer Science Volume 4537, 2007, pp 433-444

[9] Nanshan Du, Yiwen Liang, and Li Zhao. 2008. Data-flow skeleton filled with activities driven workflow design. In Proceedings of the 2nd international conference on Ubiquitous information management and communication (ICUIMC '08). ACM, New York, NY, USA, 570-574.

[10] Sun, S.X., Zhao, J.L., Nunamaker, J.F., Liu Sheng, O.R.: Formulating the Data Flow Perspective for Business Process Management. Information Systems Research 17(4), 374–391 (2006)

[11] Sundari, M.H., Sen, A.K., Bagchi, A.: Detecting Data Flow Errors in Workflows: A Systematic Graph Traversal Approach. In: 17th Workshop on Information Technology & Systems (WITS 2007), Montreal (2007)

[12] Cristina Claudia DOLEAN, Razvan PETRUSEL, Data-Flow Modeling: A Survey of Issues and Approaches. December 2012, Informatica Economica; 2012, Vol. 16 Issue 4, p117.

[13] T. Murata, “Petri Nets: Properties, Analysis and Applications,” *Proceedings of the IEEE*, vol. 77, no. 4, pp. 541–580, April, 1989.

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

[15] Cong Liu, Qingtian Zeng, Jie Zou, Faming Lu and Qingxin Wu, “Invariant Decomposition Conditions for Petri Nets Based on the Index of Transitions,” *Information Technology Journal*, vol. 11, pp. 768-774, 2012.

[16] Cong Liu, Qingtian Zeng, Hua Duan, Mengchu Zhou, Faming Lu, and Jiujun Cheng, “E-Net modeling, analysis and resource management for emergency response processes”, *IEEE Transaction on System, Man and Cybernetics: Systems*, forthcoming in 2014.

[17] Jiujun Cheng, Cong Liu, Mengchu Zhou, Qingtian Zeng, and Antti Ylä-Jääski, “Automatic Composition of Semantic Web Services Based on Fuzzy Predicate Petri Nets”, IEEE Transaction on Automation Science and Engineering, forthcoming in 2014.

[18] Qingtian Zeng, Faming Lu, Cong Liu, Hua Duan, and Changhong Zhou, “Modeling and Verification for Cross-department Collaborative Medical Business Processes Using Extended Petri Nets”, IEEE Transaction on System, Man and Cybernetics: Systems, forth coming in 2014.

[19] Hua Duan, Qingtian Zeng, Huaiqing Wang, Sherry X. Sun, Dongming Xu. Classification and Evaluation of Timed Running Logs of Workflows based on Process Mining, Journal of Systems and Software, Volume 82, Issue 3, March 2009, Pages 400-410.

[20] Sherry X. Sun, Qingtian Zeng, Huaiqing Wang, Process-Mining-Based Workflow Model Fragmentation for Distributed Execution, IEEE Transactions on Systems, Man and Cybernetics, Part A: Systems and Humans, March 2011,Volume: 41 Issue:2, On page(s): 294 – 310.

[21] Qingtian Zeng, Hua Duan. Behavior Description for Complex Flexible Manufacturing System Based on Decomposition of Petri Net, International Journal of Computer Systems Science and Engineering, Vol. 22, No.6, pp. 359-363, 2007.

[22] Qingtian Zeng. Two Symmetrical Decomposition Methods for Structure-complex Petri Net and Their Applications, Eighth ACIS International Conference on Software Engineering, Artificial Intelligence, Networking, and Parallel/Distributed Computing, 2007. SNPD 2007. Volume: 3, On page(s): 1101-1106.

[23] Qingtian Zeng, Xiaowen Hu, Jun Zhu, Hua Duan. A Polynomial-time Decomposition Algorithm for a Petri Net based on Indexes of Places, Journal of Applied Science, Volume: 8, Issue: 24, Pages: 4668-4673, 2008.

[24] Qingtian Zeng. A Polynomial-time Decomposition Algorithm for a Petri Net based on Indexes of Transitions, Information Technology Journal, 10(4):856-862. 2011.

[25] Qingtian Zeng. A Construction Method for the Process Expression of Petri Net Based on Decomposition, Information Technology Journal, Volume 7, Issue 3, Pages 420-429, 2008.

[26] Rong Hua, You Fu, Jianzhi, CongLiu, “Petri Net-based modeling and verification of automatic train speed control system”, Applied Mechanics and Materials, 2014, Vols. 571(1):395-399.

[27] Jun Zhu, Qingtian Zeng. Property analysis of Petri Nets Based on Decomposition according to indexes of Transitions [J]. Journal of Frontiers of computer science and technology, 4(08):1673-9418, 2010.

[28] Zhehui Wu. An introduction to Petri Net [M].Beijing: Publishing House of China Machine, 2006.

[29] Qingtian Zeng, Zhehui Wu. Decomposition method of Petri net based on index of places [J]. Journal of Computer Science, 29 (4): 15-17, 2002.

[30] Qingtian Zeng, Faming Lu, Cong Liu, and Decun Meng, “Modeling and Analysis for Cross-organizational Emergency Response Systems, Journal of Computers in Chinese, 2013, 36(11): 2290-2302.