International Journal of Innovative
Computing, Information and Control
Volume **8**, Number **7(B)**, July **2012**

ICIC International ©2012 ISSN 1349-4198
pp. 5217–5236

A PATTERN BASED APPROACH FOR STRUCTURAL ANALYSIS OF WORKFLOW MODELS

Chao Ou-Yang and Hsin Jung Cheng

Department of Industrial Management
National Taiwan University of Science and Technology
No. 43, Sec. 4, Keelung Rd., Taipei 106, Taiwan
ouyang@mail.ntust.edu.tw

Received March 2011; revised August 2011

Abstract. *A workflow system is normally the conventional means of supporting business process reengineering (BPR). A reference process model can be used as a TO-BE model in a BPR project. An appropriate process model should not contain structural problems. An unstructured model may cause deadlock or unfinished instances that stay in a system. Therefore, the structure of the reference model must be verified to reduce the number of errors during implementation. This work presents a novel pattern-based PN structural analytical approach. Several workflow patterns and their PN-based identification algorithms are also developed. An approach to transforming a PN model into a pattern-based process diagram (PBPD) is specified. Via the PBPD, several structure-related flaws, such as deadlock and improper nesting, are analyzed using a Split-Join Routing Table (SJRT). Analytical results support the re-design of a workflow model and enhance its feasibility. The main contribution of the proposed approach is that the PBPD can reduce the complexity of a PN model and subsequently increase analytical efficiency. An industrial reference model is used as an empirical case during the implementation phase of the proposed concepts.*

Keywords: Business process management, Petri-net, Workflow model structural analysis, Pattern-based workflow diagram

1. **Introduction.** The complexity of business processes has increased markedly due to the application of information technologies, such as Enterprise Resource Planning (ERP), and business globalization. A workflow system is usually the primary tool supporting business process reengineering (BPR) [1,2]. However, implementing a workflow system typically requires a reference model as the TO-BE model in a BPR project. This model can be from software providers, such as the SAP "best practice models", benchmarking companies, and consortium partners, or constructed by system developers based on their industrial experience.

An appropriate process model should not contain structural problems. A structured workflow model is a model in which each of its split control elements (AND, OR) matches a joint control element [3]. Certain process flaws, including deadlock and multiple instances (unfinished instances in a process), may occur in an unstructured process. Therefore, workflow structural analysis is a necessary stage when developing and implementing a workflow system.

Petri Nets (PNs) have been commonly used while modeling and analyzing business processes. However, due to the increasing complexity of workflow systems, a PN model may be difficult to interpret, let alone analyzing structural flows of a model. Therefore, this work proposes a pattern-based PN structure analytical approach. Several workflow patterns and their PN-based identification algorithms are first defined. Then, based on recognized patterns, a novel approach to transforming a PN model into a pattern-based process diagram (PBPD) is utilized. A Split-Join Routing Table (SJRT) is created based on the PBPD. A novel analytical approach is employed to analyze the casual relationships among identified split and join patterns and to identify structural problems such as deadlock, improper corresponding structure and improper nesting. Analytical results can be utilized to assist in re-engineering a workflow reference model.

For a BPR project, structural analysis of a workflow reference model should be conducted before employing that model as a TO-BE model. Then, the novel pattern-based PN workflow model structural analysis approach is applied to supporting a BPR project. The proposed concept also supports the synchronization of a BPR process. That is, a reference model structure is analyzed and redesigned during the TO-BE model construction stage and hence can limit the logical errors of the business process occurred after the implementation.

Finally, the proposed concepts were implemented by integrating a PN modeling tool and a developed analytical environment. An industrial reference process, "acquisition handling" from ARIS "mechanical engineering reference model" was selected as an empirical test case.

2. Literature Review.

A workflow pattern is the primary control element embedded in a workflow model. Aalst et al. characterized four workflow patterns corresponding to routing constructs encountered when modeling and analyzing workflows [4]. In another study, they further classified patterns into six categories that included 20 types such as Basic Control Flow Patterns, Advanced Branching and Synchronization Patterns, Structural Patterns, Patterns involving Multiple Instances, Stated-based Patterns, and Cancellation Patterns [5]. Gaaloul et al. integrated process mining and workflow patterns and proposed a pattern mining algorithm that uses statistical analysis to find "local" workflow patterns [6]. Zhou et al. proposed a method that simplifies the specification of patterns in terms of classical P/T nets [7]. In addition, the patterns are described by an algebra method, synchronic distance, based the P/T nets specification. The proposed approach has been used to verify Aalst's workflow patterns representation method and achieved similar effects.

Most workflow languages can be applied to express process routing patterns such as sequence iteration, split, and join. However, to execute a process smoothly, one must ensure that structures of processes do not conflict. A feasible process model has no structural defects. A workflow is considered well behaved when it does not produce deadlocks or allow multiple instances of the same activity [8]. Liu et al. defined a structured workflow as one in which each split control element is matched with a join control element of the same type, and these split-join pairs should be appropriately nested [3]. They also addressed problems associated with improper nesting and mismatched pairs and their behaviors.

Several researchers have developed PN schema to represent certain features of business processes. For example, Choi et al. proposed a colored PN called "Test Net" to represent AND/OR/Join/Split concepts proposed by the Workflow Management Coalition (WfMC) [9]. Buhler et al. utilized Business Process Execution Language for Web-Service (BPEL4WS) to model the workflow in an agent-based system and then transformed the model into a PN for subsequent analysis [10]. Several researchers have focused on issues associated with workflow structural changes. For instance, Sun et al. proposed an algorithm that identifies the minimal change region, and proves that the change region can be used to assess the compatibility of workflow changes [11]. Chaparro-Baquero et al. described the workflow modeling concept for the digital publishing business processes using Workflow nets and Generalized Stochastic PNs (GSPNs) to measure dependability attributes in a quantitative form [12]. Moreover, PN is also a widely used tool which can represent a discrete event system. For instance, Maeno et al. introduced an approximation method that uses PN to model the semiconductor manufacturing system and achieved efficient improvement in solving the optimal firing sequence problem via decomposing the PN [13].

Several graph reduction methods have been developed to identify structural conflicts in workflow models. For instance, Sadiq et al. defined five reduction rules to identify structural problems, such as deadlock and lack of synchronization [14]. Aalst et al. proposed an alternative algorithm that uses reduction rules to detect the structural conflicts and to transform workflow graphs into WF-nets [15]. Lee et al. introduced an alternative reduction method for generalized PN by reducing places, transitions while preserving the properties such as liveness and boundedness [16]. Sloan et al. described a set of notions which are equivalent with time Petri nets that guarantees crucial timing and concurrency properties can be preserved [17]. Xia proposed a PN based Representation for Embedded Systems (PRES+) containing a set of reduction rules to remove or merge places and transitions and to preserve total-equivalent between original PRES+ nets and reduced nets [18]. In another study, a structuring matrix (WfM) procedure was developed to group workflow activities into a structured form according to data links among activities. The structured model was then applied to re-engineering a design process [19]. Another study developed a reduction approach that simplifies the Event Process Chain (EPC) reference models and then transforms the simplified models into PNs. Invariant and human judgments have been applied during state-space analysis to verify SAP reference models [20].

In conclusion, these studies indicate that PN is an effective workflow modeling and analysis tool. Workflow patterns typically function as flow control elements in workflow models. Additionally, an appropriate workflow model should not contain structural problems. Therefore, in this work, a novel PN-pattern-based approach is utilized to support for workflow model structure analysis and to assist the re-design of workflow models.

3. **Proposed Approach.** This section specifies a novel approach that identifies workflow patterns from a PN model and then analyzes the respective workflow structure. Six workflow patterns are defined according to the causal relationships among input/output places and their respective transitions. Then, algorithms for discovering these six patterns from a PN are proposed. A PBPD is constructed according to discovered patterns. Finally, structural analysis is carried out to identify several improper workflow structures embedded in a PN model. Figure 1 shows the framework of the proposed approach.

3.1. **Identifying workflow patterns.** The principal aim of this section is to define six workflow patterns – AND-Split, AND-Join, XOR-Split, XOR-Join, Basic Building Block, and Sequence – according to the causal relationships among transitions and places. Since the algorithms for pattern identification are based on PNs, the definition and related concepts of PN are introduced as follows:

Definition 3.1. *Petri net [21]*

A Petri Net is a tuple (P, T, F, W) with

- *P is a finite set of places,*
- *T is a finite set of transitions such that $P \cap T = \phi$,*
- *$F \subseteq (P \times T) \cup (T \times P)$ is a set of arcs (flow relation), and*
- *$W : F \rightarrow \mathrm{N}$ is a function defining the weight of each arc of the net, where N is the set of natural numbers, such that $w(t, p)$ is the weight of the arc connecting t to p. Similarly, $w(p, t)$ is the weight of the arc connecting p to t.*

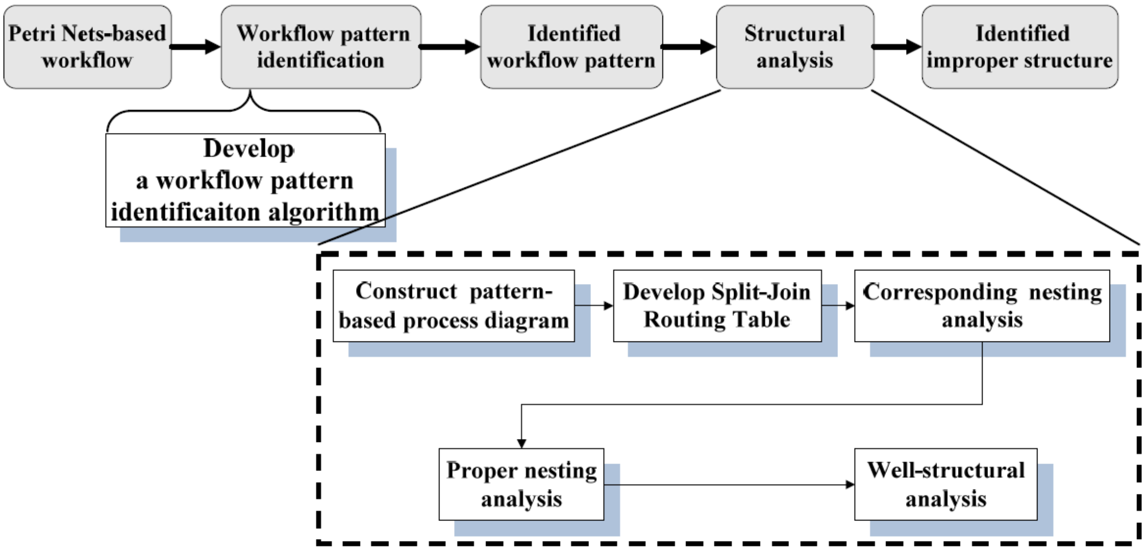


Figure 1. Framework of workflow pattern identification and process structural analysis

- *A place $p \in P$ is an input place of a transition $t \in T$ if and only if there exists a directed arc from p to t, i.e., if and only if $(p, t) \in F$. The set of input places for a transition t is denoted $\bullet t$.*
- *A place $p \in P$ is an output place of a transition $t \in T$ if and only if there exists a directed arc from t to p, i.e., if and only if $(t, p) \in F$. The set of output places for a transition t is denoted $t\bullet$.*
- *$p\bullet$ and $\bullet p$ denote the sets of transitions that share p as input places and output places, respectively.*

Additionally, $|\bullet t|$ is the number of input places for a transition t, and $|\bullet p|$ is the number of input transitions for a place p. Similarly, $|t\bullet|$ and $|p\bullet|$ are the number of output places and transitions for a transition t and a place p, respectively. Based on these definitions, let $S = (\bullet t, t\bullet, \bullet p, p\bullet)$ represent the input/output (I/O) structure of places and transitions contained in a PN model. Figures 2(1) and 2(2) show a PN model and its I/O structure.

According to the defined PN I/O structure, the related pattern set can be defined as:

Definition 3.2. *Pattern set*

A pattern set $R = \{r_{x,i}\}$ contained all of the patterns in a PN model such that each $r_{x,i} = \{[p], [t]\}$ represents an individual pattern, where $x \in \{as, aj, xs, xj, b, s\}$, and i is the sequence code.

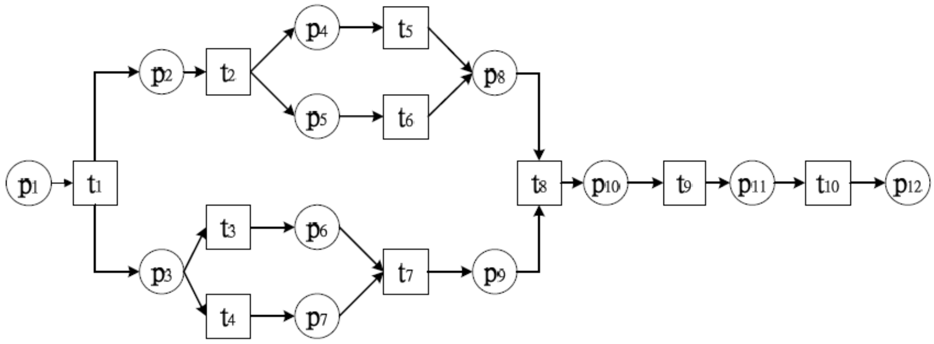
In the above definition, $r_{as,i}$ is the ith AND-Split pattern in R. Similarly, $r_{aj,i}$, $r_{xs,i}$, $r_{xj,i}$, $r_{b,i}$ and $r_{s,i}$ specify the ith AND-Join pattern, XOR-Split pattern, XOR-Join pattern, Basic Building Block pattern, and Sequence pattern, respectively.

For instance, $R = \{r_{as,1}, r_{as,2}, r_{aj,1}, r_{aj,2}, r_{xs,1}, r_{xj,1}, r_{b,1}, r_{b,2}, r_{s,1}\}$ indicates that a PN model has two AND-Split patterns, two AND-Join patterns, one XOR-Split pattern, one XOR-Join pattern, two Basic Building Block patterns, and one Sequence pattern.

The next step is to find all patterns in a PN model according to the causal relationships embedded in the I/O structure of the model. The algorithm to find an AND-Split pattern is as follows.

① $t^i_{as} = \{t \in T | |\bullet t| = 1 \wedge |t\bullet| \geq 2\}$, $p^h_{as,i} = \{\bullet t^i_{as} | \bullet t^i_{as} \in P\}$, $p^t_{as,i} = \{t^i_{as} \bullet | t^i_{as} \bullet \in P\}$.

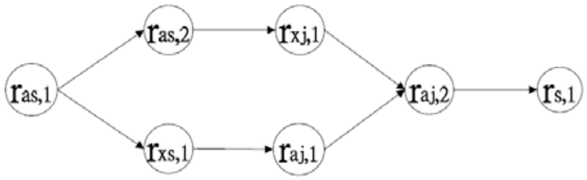
② $r_{as,i} = \left\{\left[p^h_{as,i}, p^t_{as,i}\right], \left[t^i_{as}\right]\right\}$, *where i is a sequence code.*



(1)

$$S\left\{\begin{array}{lll} \bullet t_1=\{p_1\},\ t_1\bullet=\{p_2,p_3\} & \bullet t_2=\{p_2\},\ t_2\bullet=\{p_4,p_5\} & \bullet t_3=\{p_3\},\ t_3\bullet=\{p_6\} \\ \bullet t_4=\{p_3\},\ t_4\bullet=\{p_7\} & \bullet t_5=\{p_4\},\ t_5\bullet=\{p_8\} & \bullet t_6=\{p_5\},\ t_6\bullet=\{p_8\} \\ \bullet t_7=\{p_6,p_7\},\ t_7\bullet=\{p_9\} & \bullet t_8=\{p_8,p_9\},\ t_8\bullet=\{p_{10}\} & \bullet t_9=\{p_{10}\},\ t_9\bullet=\{p_{11}\} \\ \bullet t_{10}=\{p_{11}\},\ t_{10}\bullet=\{p_{12}\} & & \end{array}\right\}$$

(2)



(3)

FIGURE 2. (1) An original PN model; (2) its input/output structure (continued); (3) its pattern-based process diagram (continued)

The principal aim of this novel algorithm is to find a transition t_{as}^{i} from the I/O structure of a PN model that has one input place and at least two output places, i.e., $|\bullet t| = 1$ and $|t\bullet| \geq 2$. In the algorithm, $p_{as,i}^{h}$ is the set containing the input place and $p_{as,i}^{t}$ specifies the set, including output places. Finally, the transition, input place and output places are formed as an individual pattern set $r_{as,i} = \left\{ \left[p_{as,i}^{h}, p_{as,i}^{t}\right], \left[t_{as}^{i}\right]\right\}$.

For instance, from the I/O structure (Figure 2(2)), t_1 and t_2 are transitions that satisfy the condition $t_{as}^{i} = \{t \in T|\, |\bullet t| = 1 \wedge |t\bullet| \geq 2\}$ because $\bullet t_1 = p_1$ and $t_1\bullet = p_2, p_3$ are the input/output places for t_1, and $\bullet t_2 = p_2$ and $t_2\bullet = p_4, p_5$ are the input/output places for t_2. Therefore, $r_{as,1} = \{[p_1, p_2, p_3], [t_1]\}$ and $r_{as,2} = \{[p_2, p_4, p_5], [t_2]\}$ are formed as two subsets in $R = \{r_{as,1}, r_{as,2}\}$ symbolizing the two AND-Split patterns.

The search algorithms for the other three patterns – AND-Join, XOR-Split and XOR-Join are the same as the AND-Split pattern. That is, the first stage is to identify transitions, such that the number of their input and output places should satisfy the related constraints. Then, the causal relationship among input/output places and the transition must be identified.

In addition to the four AND\XOR patterns, two other patterns types, Basic Build Block and Sequence, are considered. A Basic Build Block pattern is typically used to connect two patterns. Basically, the transition with one input and one output place is grouped as one Basic Build Block pattern.

To identify the sequence pattern, the transitions with a single input and single output are identified first. Then, among the recognized transitions, if a set of transitions exist, such that the output places of one transition are shared by the input places of another transition, these transitions and places will be grouped as one sequence pattern set. However, two transitions that share an input place or output place should be excluded from the set.

3.2. **PN-based process structural analysis.**

In this section, the soundness of a process model is determined using a novel approach. A PBPD is first constructed first based on the causal relationships among the identified patterns. Then, by examining the connected routes between split and join patterns, an SJRT is created. Finally, various improper workflow structures are identified based on routing information in the SJRT.

3.2.1. *Construct a pattern-based process diagram.*

The first stage in process structure analysis is to construct a PBPD. The main objective is to create a concise process model while maintaining the routing structure of the original PN model. This diagram represents the connections among identified patterns. By applying this diagram, the task of process structural analysis can be simplified. The PBPD is defined as follows.

Definition 3.3. *Pattern-based process diagram (PBPD)*
A PBPD is a dual (R, A) *where*

- $R = \{r_{x,i}\}$ *is a finite set of workflow patterns, where* $x \in \{as, aj, xs, xj, b, s\}$ *and* i *indicate the sequence code.*
- $\forall_{r_{x,ia}, r_{x,ib} \in r_{x,i}} A \subseteq (r_{x,ia} \times r_{x,ib})$ *is a set of arcs (pattern flow relations), which is connected from pattern* $r_{x,ia}$ *to pattern* $r_{x,ib}$*.*

This definition means that a PBPD comprises a workflow pattern set R and a pattern flow relational set A, indicating that arcs connect a split and a join pattern. A PN model (Figure 2(1)) can be mapped onto its PBPD (Figure 2(3)).

3.2.2. *Develop a split-join routing table (SJRT).*

In this section, a novel approach that applies the SJRT to identifying improper structures is proposed. An SJRT represents routes that can be identified from a PBPD and connects split patterns and join patterns. By applying this information, the matched pattern pairs embedded in a PBPD can be located. An SJRT is defined as follows.

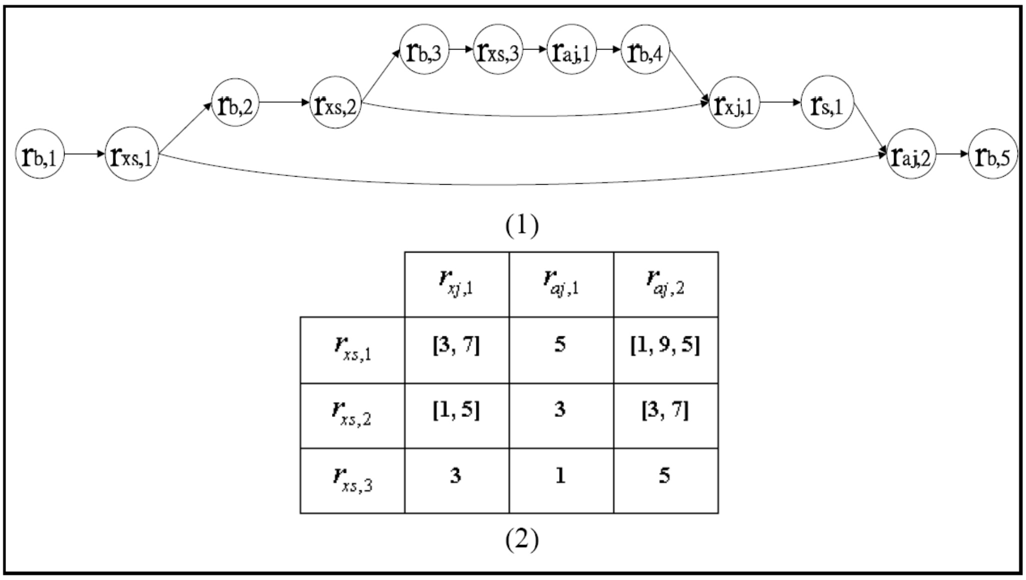
Definition 3.4. *Split-Join Routing Table (SJRT)*
An SJRT is a triple $SJRT = (R_{sp}, C_{jp}, a_{ij})$*, where*

- $R_{sp} = [r_{sp,i}]$ *is a set of split pattern indices, where* i *is the number of rows and* $sp \in \{as, xs\}$.
- $C_{jp} = [r_{jp,j}]$ *is a set of join pattern indices, where* j *is the number of columns and* $jp \in \{aj, xj\}$.
- $a_{ij} = \left[\left|a_{ij}^{x}\right|\right]$ *is a routing set in row* i*; column* j *of the table where* a_{ij}^{x} *is the set of arcs in a route connecting a split pattern* $r_{sp,i}$ *and join pattern* $r_{jp,j}$*, and* x *is a sequence code.*

$\left|a_{ij}^{x}\right|$ *is the number of arcs in* a_{ij}^{x}*; that is, the number of arcs in a route connecting a split pattern* $r_{sp,i}$ *and a join pattern* $r_{jp,j}$*.*
Additionally, $|a_{ij}|$ *is the number of elements in* a_{ij}*.*

Thus, the principal task in building a SJRT is to identify the number of arcs in routes linking a split pattern and a join pattern in a PBPD.

Figure 3 shows a sample PBPD and its SJRT. Two paths connect the XOR-Split pattern $r_{xs,1}$ and the XOR-Join pattern $r_{xj,1}$. Among these paths, one contained three links –



(1)

	$r_{xj,1}$	$r_{aj,1}$	$r_{aj,2}$
$r_{xs,1}$	[3, 7]	5	[1, 9, 5]
$r_{xs,2}$	[1, 5]	3	[3, 7]
$r_{xs,3}$	3	1	5

(2)

FIGURE 3. Example of a sample PBPD and its SJRT

$r_{xs,1} \to r_{b,2}$, $r_{b,2} \to r_{xs,2}$, and $r_{xs,2} \to r_{xj,1}$ – and another path has seven links. Therefore, $a_{11}^1 = [r_{xs,1} \to r_{b,2}, r_{b,2} \to r_{xs,2}, r_{xs,2} \to r_{xj,1}]$, $a_{11}^2 = [r_{xs,1} \to r_{b,2}, r_{b,2} \to r_{xs,2}, r_{xs,2} \to r_{b,3}, r_{b,3} \to r_{xs,3}, r_{xs,3} \to r_{aj,1}, r_{aj,1} \to r_{b,4}, r_{b,4} \to r_{xj,1}]$, and $a_{11} = [|a_{11}^x|] = [|a_{11}^1|, |a_{11}^2|] = [3, 7]$, $|a_{11}| = 2$.

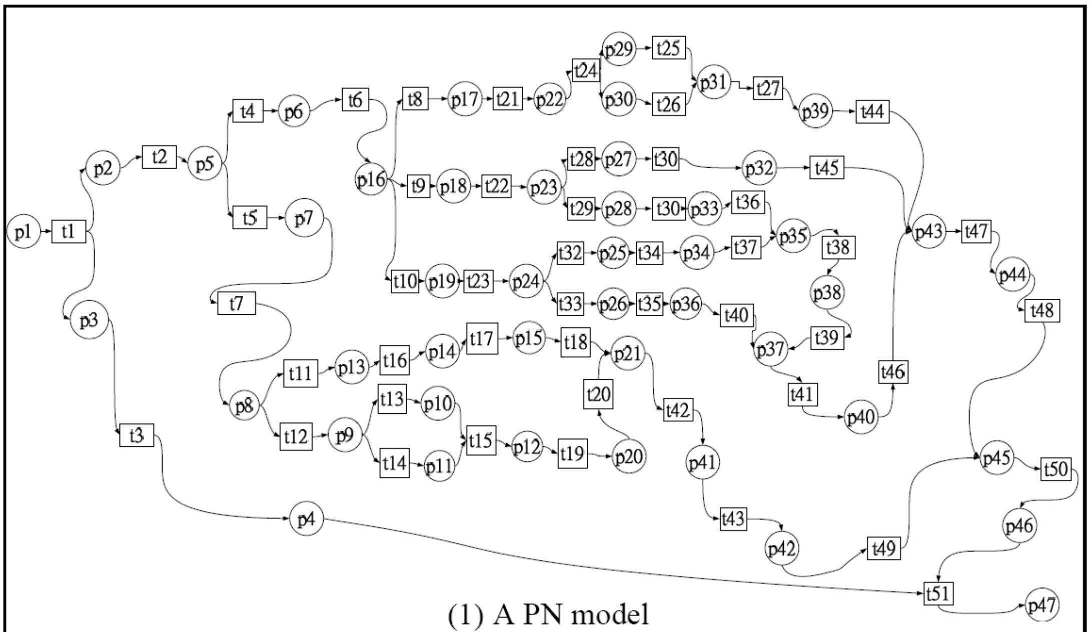
3.2.3. *Identification of matched pairs.* Once an SJRT is constructed, the next task is to identify matched split-join pattern pairs from the SJRT. The main aim is to identify a split-join pattern pair based on the number of paths connecting these patterns and number of arcs in each route. The PN model and its PBPD (Figure 4) and Figures 5-9 are used to describe the proposed algorithm.

The procedure contains two main stages. The first stage is to identifying the rows in an SJRT, such that at least one element in each row is a single-value element. The second stage focuses on analyzing the rows in SJRT, such that all elements in a row are multiple-value elements. The procedure is implemented iteratively between the two stages according to the status of the modified SJRT.

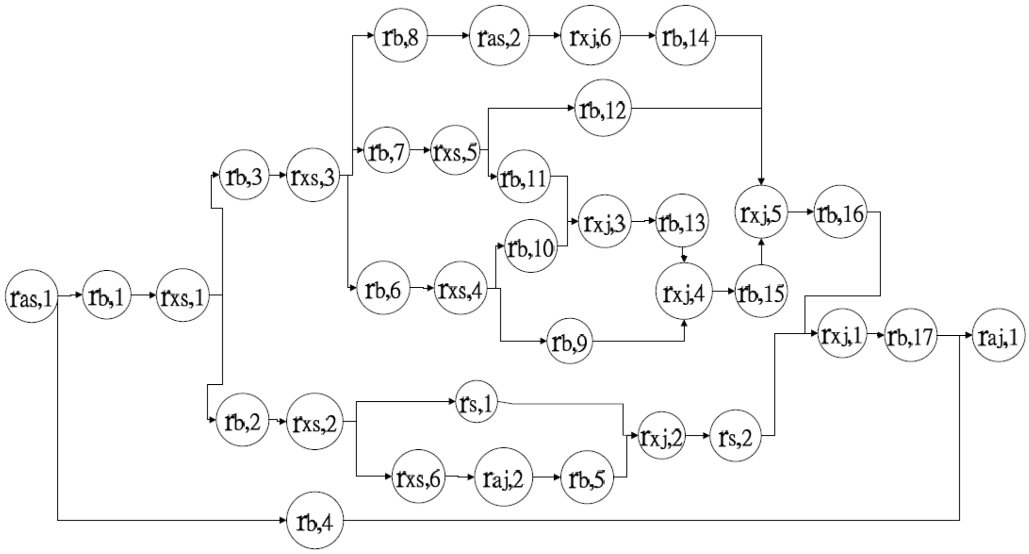
The algorithm for analyzing rows, such that all components are single-value elements is:

```
Let $USJRT = SJRT$
1. For $i = 1..n$, find row $i$ in USJRT, such tat $\forall j \to |a_{ij}| = 1$
2. Let $BM = [a_{ij}^x]$
3. $\forall |a_{ij}^x| \neq 0$, find $\min |a_{ij}^x|$ from BM and let the corresponding $r_{sp,i}$ and $r_{jp,j}$ formed a
   matched pair $[r_{sp,i}, r_{jp,j}]$
4. Let $USJRT = USJRT - [r_{sp,i}, r_{jp,j}, a_{ij}]$
5. Back to Step-1 until BM becomes an empty matrix
```

In the initial stage, an updated SJRT (USJRT) is constructed (Figure 5(1)). Then, the first step selects row(s) such that all elements in each row contain only one value. The selected routing sets form a buffer matrix (BM) (Step 2) (Figure 5(2)). The next task is to find the element with a non-zero minimum value (a_{18}^1 and a_{22}^1), and the corresponding split and join patterns are then allowed to form a matched pair $[r_{as,2}, r_{xj,6}]$ (Step 3). The corresponding row and column of this pair are removed from the USJRT (Step 4) (Figure 6(3)). These steps are repeated until all single-value rows are removed (Step 5). So, a



(1) A PN model



(2) A PBPD

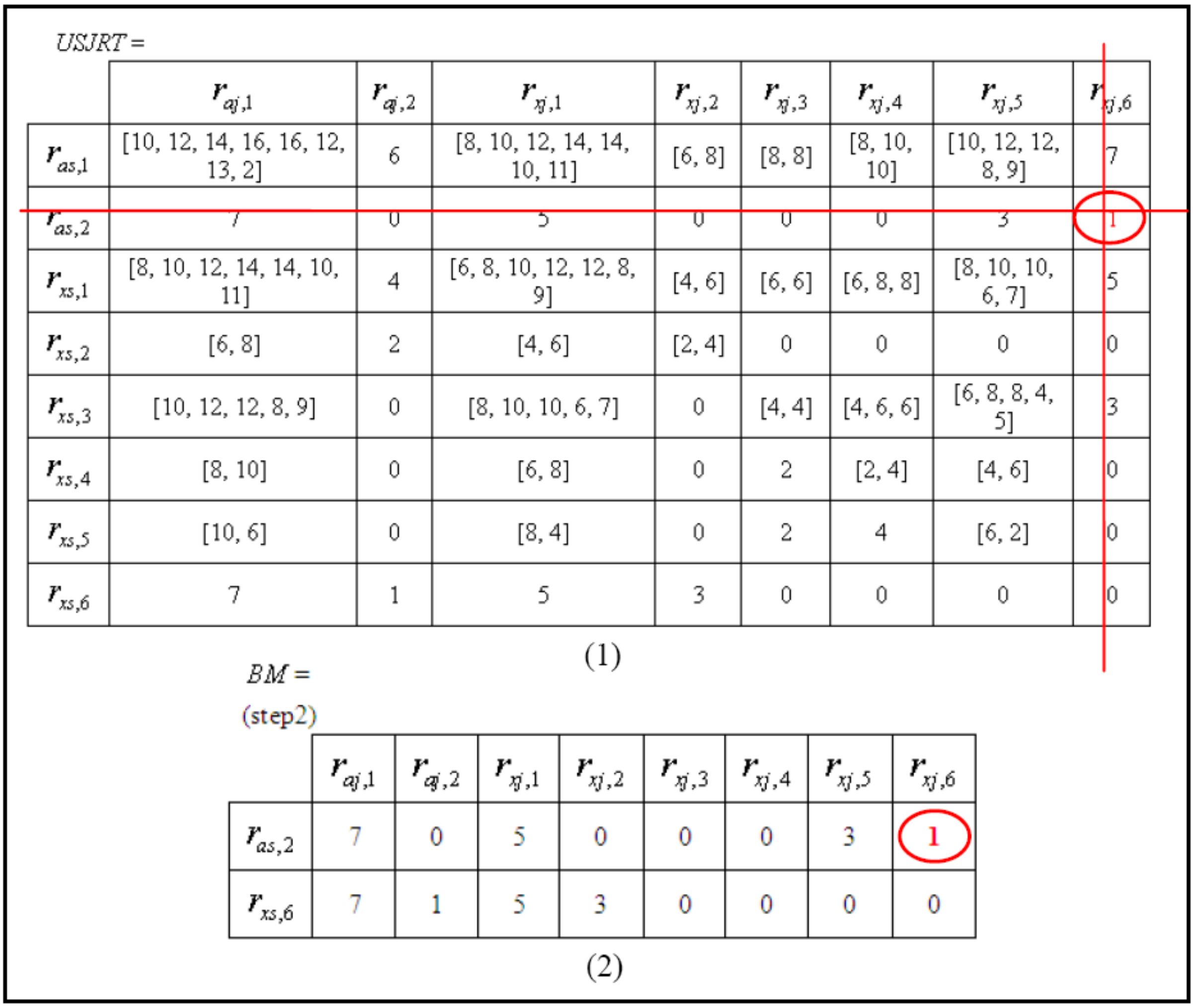
FIGURE 4. Example of a PN model and a PBPD for identifying matched pairs

new matched pattern pair $[r_{xs,6}, r_{aj,2}]$ is identified (Figure 6(4)) and removed from USJRT (Figure 7(5)).

The main goal of these steps is to locate the split-join pattern pair, such that only one route connecting the pair. The pair with the minimum number of arcs in the route is then selected. Conversely, among the single route split-join pattern pairs, the pair with the shortest path is selected first.

Once all rows with single values have been analyzed, the next step is to analyze the rows containing both single and multiple-values elements. The analytical algorithm is as follows:

6. *For* $i = 1..n$*, find the row* i *in USJRT, such that at least one element* $j \rightarrow |a_{ij}| = 1$
7. *Let* $BM = \left[a_{ij}^{x}\right]$



USJRT =

	$r_{aj,1}$	$r_{aj,2}$	$r_{xj,1}$	$r_{xj,2}$	$r_{xj,3}$	$r_{xj,4}$	$r_{xj,5}$	$r_{xj,6}$
$r_{as,1}$	[10, 12, 14, 16, 16, 12, 13, 2]	6	[8, 10, 12, 14, 14, 10, 11]	[6, 8]	[8, 8]	[8, 10, 10]	[10, 12, 12, 8, 9]	7
$r_{as,2}$	7	0	5	0	0	0	3	1
$r_{xs,1}$	[8, 10, 12, 14, 14, 10, 11]	4	[6, 8, 10, 12, 12, 8, 9]	[4, 6]	[6, 6]	[6, 8, 8]	[8, 10, 10, 6, 7]	5
$r_{xs,2}$	[6, 8]	2	[4, 6]	[2, 4]	0	0	0	0
$r_{xs,3}$	[10, 12, 12, 8, 9]	0	[8, 10, 10, 6, 7]	0	[4, 4]	[4, 6, 6]	[6, 8, 8, 4, 5]	3
$r_{xs,4}$	[8, 10]	0	[6, 8]	0	2	[2, 4]	[4, 6]	0
$r_{xs,5}$	[10, 6]	0	[8, 4]	0	2	4	[6, 2]	0
$r_{xs,6}$	7	1	5	3	0	0	0	0

(1)

BM =

(step2)

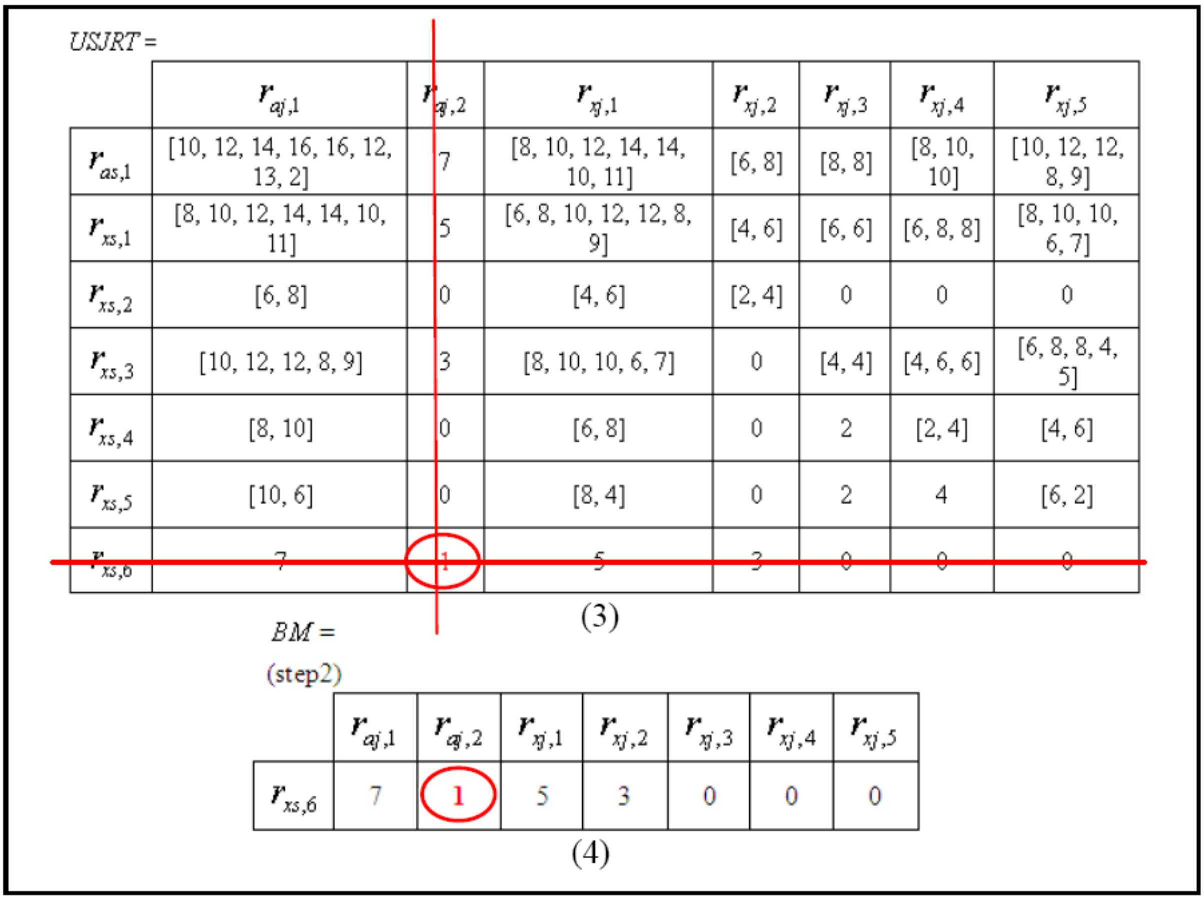
	$r_{aj,1}$	$r_{aj,2}$	$r_{xj,1}$	$r_{xj,2}$	$r_{xj,3}$	$r_{xj,4}$	$r_{xj,5}$	$r_{xj,6}$
$r_{as,2}$	7	0	5	0	0	0	3	1
$r_{xs,6}$	7	1	5	3	0	0	0	0

(2)

FIGURE 5. Procedure for identifying matched pairs

8. *Reorder the row in BM such that* $||a_{1j}| \neq 1|_{j=1}^{m} \leq ||a_{2j}| \neq 1|_{j=1}^{m} \leq \ldots ||a_{nj}| \neq 1|_{j=1}^{m}$ *and* $\min\left|a_{1j}^{x}\right| \leq \min\left|a_{2j}^{x}\right| \leq \ldots \min\left|a_{nj}^{x}\right|$, *where*
   $||a_{ij}| \neq 1|_{j=1}^{m}$: *number of elements in row i such that* $|a_{ij}| \neq 1$
   $\min\left|a_{ij}^{x}\right|$: *the minimum route values in row i of BM*
9. $\forall\, ||a_{1j}| \neq 1|$, *find* $\min\left|a_{1j}^{x}\right|$ *and let the corresponding* $r_{sp,i}$ *and* $r_{jp,j}$ *form a matched pair* $[r_{sp,i}, r_{jp,j}]$
10. *For* a_{1j}^{x} *in BM, find the respective* a_{ij} *in USJRT*
11. *Let* $USJRT = USJRT - [r_{sp,i}, r_{jp,j}, a_{ij}]$
12. *Back to Step-6 until BM becomes an empty matrix*

The first step is to identify qualified rows and construct the BM (Steps 6 and 7) (Figures 7(5) and 7(6)). Then, the rows in the BM are reordered in an ascending order, such that the row containing the fewest multiple-value elements is in the first row. If two rows have the same number of multiple value elements, then the row containing the smallest route value is in the first row (Step 8) (Figure 7(7)).



USJRT =

	$r_{aj,1}$	$r_{aj,2}$	$r_{xj,1}$	$r_{xj,2}$	$r_{xj,3}$	$r_{xj,4}$	$r_{xj,5}$
$r_{as,1}$	[10, 12, 14, 16, 16, 12, 13, 2]	7	[8, 10, 12, 14, 14, 10, 11]	[6, 8]	[8, 8]	[8, 10, 10]	[10, 12, 12, 8, 9]
$r_{xs,1}$	[8, 10, 12, 14, 14, 10, 11]	5	[6, 8, 10, 12, 12, 8, 9]	[4, 6]	[6, 6]	[6, 8, 8]	[8, 10, 10, 6, 7]
$r_{xs,2}$	[6, 8]	0	[4, 6]	[2, 4]	0	0	0
$r_{xs,3}$	[10, 12, 12, 8, 9]	3	[8, 10, 10, 6, 7]	0	[4, 4]	[4, 6, 6]	[6, 8, 8, 4, 5]
$r_{xs,4}$	[8, 10]	0	[6, 8]	0	2	[2, 4]	[4, 6]
$r_{xs,5}$	[10, 6]	0	[8, 4]	0	2	4	[6, 2]
~~$r_{xs,6}$~~	~~7~~	~~1~~	~~5~~	~~3~~	~~0~~	~~0~~	~~0~~

(3)

BM =
(step2)

	$r_{aj,1}$	$r_{aj,2}$	$r_{xj,1}$	$r_{xj,2}$	$r_{xj,3}$	$r_{xj,4}$	$r_{xj,5}$
$r_{xs,6}$	7	1	5	3	0	0	0

(4)

FIGURE 6. Procedure for identifying matched pairs (continued)

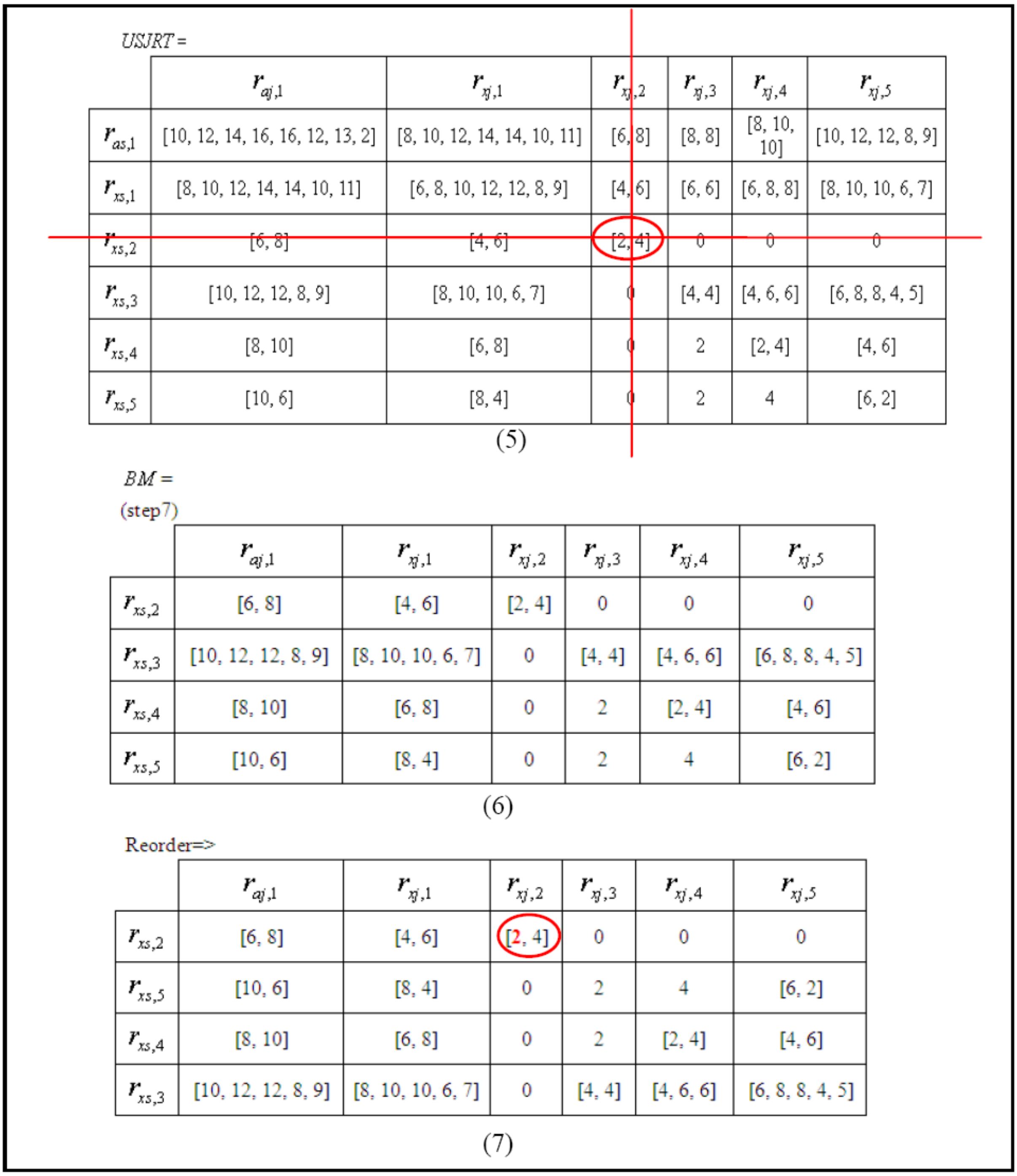
For example, Figure 7(6) shows the identified rows containing both single and multiple values. The number of multiple-value elements in each row is as follows:

$$\begin{aligned}
&\left|\left|a_{1j}\right| \neq 1\right|_{j=1}^{m} = 3 \quad (|a_{11}| = 2, |a_{12}| = 2, |a_{13}| = 2),\\
&\left|\left|a_{2j}\right| \neq 1\right|_{j=1}^{m} = 5 \quad (|a_{21}| = 5, |a_{22}| = 5, |a_{24}| = 2, |a_{25}| = 3, |a_{26}| = 5),\\
&\left|\left|a_{3j}\right| \neq 1\right|_{j=1}^{m} = 4 \quad (|a_{31}| = 2, |a_{32}| = 2, |a_{35}| = 2, |a_{36}| = 2),\\
&\left|\left|a_{4j}\right| \neq 1\right|_{j=1}^{m} = 3 \quad (|a_{41}| = 2, |a_{42}| = 2, |a_{46}| = 2).
\end{aligned}$$

Thus, rows 1 and 4 contain three multiple-value elements. Since the minimum route values are the same in these rows ($|a_{13}^1| = |a_{46}^2| = 2$), they are put in the first and second rows of the reordered BM. Therefore, based on the ascending order, the element in row 4 is switched with elements in row 2. Figure 7(7) shows the reordered BM.

Then, the element that contains the smallest value is identified (Step 9) ($\min \left|a_{1j}^x\right| = |a_{13}^1| = 2$). The corresponding split and join patterns of that element then form a pattern pair ($[r_{xs,2}, r_{xj,2}]$) and removed from USJRT (Steps 10 and 11) (Figure 8(8)). Since there are two rows which still contain multi-value elements, the process is carried out again (Figures 8(8)-8(12)) and the fourth and fifth pairs, $[r_{xs,5}, r_{xj,5}]$ and $[r_{xs,4}, r_{xj,4}]$, are identified. Figure 9(13) shows the updated USJRT at this stage.

The goal in selecting multi-value instead of single-value elements is that when several join patterns correspond to a split pattern, the single-value route must be included in a multi-value path and, hence, should not be considered. For example, in the USJRT (Figure 5(1)), row $r_{xs,5}$ is a mixed-values row. The two single-value routes, $r_{xs,5} \rightarrow r_{xj,3}$



USJRT =

	$r_{aj,1}$	$r_{xj,1}$	$r_{xj,2}$	$r_{xj,3}$	$r_{xj,4}$	$r_{xj,5}$
$r_{as,1}$	[10, 12, 14, 16, 16, 12, 13, 2]	[8, 10, 12, 14, 14, 10, 11]	[6, 8]	[8, 8]	[8, 10, 10]	[10, 12, 12, 8, 9]
$r_{xs,1}$	[8, 10, 12, 14, 14, 10, 11]	[6, 8, 10, 12, 12, 8, 9]	[4, 6]	[6, 6]	[6, 8, 8]	[8, 10, 10, 6, 7]
$r_{xs,2}$	[6, 8]	[4, 6]	[2, 4]	0	0	0
$r_{xs,3}$	[10, 12, 12, 8, 9]	[8, 10, 10, 6, 7]	0	[4, 4]	[4, 6, 6]	[6, 8, 8, 4, 5]
$r_{xs,4}$	[8, 10]	[6, 8]	0	2	[2, 4]	[4, 6]
$r_{xs,5}$	[10, 6]	[8, 4]	0	2	4	[6, 2]

(5)

BM =
(step7)

	$r_{aj,1}$	$r_{xj,1}$	$r_{xj,2}$	$r_{xj,3}$	$r_{xj,4}$	$r_{xj,5}$
$r_{xs,2}$	[6, 8]	[4, 6]	[2, 4]	0	0	0
$r_{xs,3}$	[10, 12, 12, 8, 9]	[8, 10, 10, 6, 7]	0	[4, 4]	[4, 6, 6]	[6, 8, 8, 4, 5]
$r_{xs,4}$	[8, 10]	[6, 8]	0	2	[2, 4]	[4, 6]
$r_{xs,5}$	[10, 6]	[8, 4]	0	2	4	[6, 2]

(6)

Reorder=>

	$r_{aj,1}$	$r_{xj,1}$	$r_{xj,2}$	$r_{xj,3}$	$r_{xj,4}$	$r_{xj,5}$
$r_{xs,2}$	[6, 8]	[4, 6]	[2, 4]	0	0	0
$r_{xs,5}$	[10, 6]	[8, 4]	0	2	4	[6, 2]
$r_{xs,4}$	[8, 10]	[6, 8]	0	2	[2, 4]	[4, 6]
$r_{xs,3}$	[10, 12, 12, 8, 9]	[8, 10, 10, 6, 7]	0	[4, 4]	[4, 6, 6]	[6, 8, 8, 4, 5]

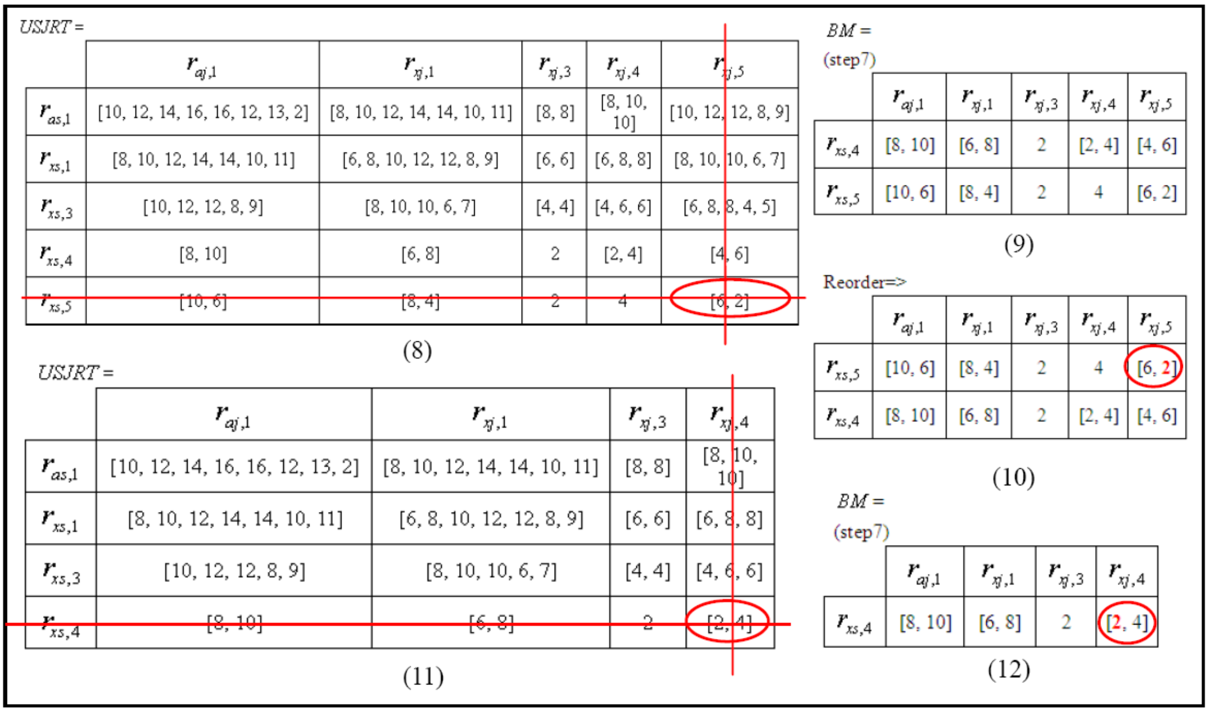
(7)

FIGURE 7. Procedure for identifying matched pairs (continued)

and $r_{xs,5} \to r_{xj,4}$, are included in multi-value routes $r_{xs,5} \to r_{aj,1}$, $r_{xs,5} \to r_{xj,1}$ and $r_{xs,5} \to r_{xj,5}$. Therefore, the pair with the fewest arcs in multi-value routes (the shortest path) is selected; that is, $[r_{xs,5}, r_{xj,5}]$ in this case.

The next stage is to identify the patterns from rows, such that all contained constructs are multi-value elements. This stage has two tasks. The first task is to identify the pattern pairs, such that for each split pattern, the routes linked to its join pattern contain the same number of arcs. The algorithm is as follows:

13. *For* $i = 1..n$, *find the row* i *in USJRT, such that* $\forall a_{ij}$ *in that row,* $||a_{i1}| \neq 1| = ||a_{i2}| \neq 1| = \ldots ||a_{nj}| \neq 1|$



USJRT =

	$r_{aj,1}$	$r_{xj,1}$	$r_{xj,3}$	$r_{xj,4}$	$r_{xj,5}$
$r_{as,1}$	[10, 12, 14, 16, 16, 12, 13, 2]	[8, 10, 12, 14, 14, 10, 11]	[8, 8]	[8, 10, 10]	[10, 12, 12, 8, 9]
$r_{xs,1}$	[8, 10, 12, 14, 14, 10, 11]	[6, 8, 10, 12, 12, 8, 9]	[6, 6]	[6, 8, 8]	[8, 10, 10, 6, 7]
$r_{xs,3}$	[10, 12, 12, 8, 9]	[8, 10, 10, 6, 7]	[4, 4]	[4, 6, 6]	[6, 8, 8, 4, 5]
$r_{xs,4}$	[8, 10]	[6, 8]	2	[2, 4]	[4, 6]
~~$r_{xs,5}$~~	~~[10, 6]~~	~~[8, 4]~~	~~2~~	~~4~~	~~[6, 2]~~

(8)

BM =
(step7)

	$r_{aj,1}$	$r_{xj,1}$	$r_{xj,3}$	$r_{xj,4}$	$r_{xj,5}$
$r_{xs,4}$	[8, 10]	[6, 8]	2	[2, 4]	[4, 6]
$r_{xs,5}$	[10, 6]	[8, 4]	2	4	[6, 2]

(9)

Reorder=>

	$r_{aj,1}$	$r_{xj,1}$	$r_{xj,3}$	$r_{xj,4}$	$r_{xj,5}$
$r_{xs,5}$	[10, 6]	[8, 4]	2	4	[6, 2]
$r_{xs,4}$	[8, 10]	[6, 8]	2	[2, 4]	[4, 6]

(10)

USJRT =

	$r_{aj,1}$	$r_{xj,1}$	$r_{xj,3}$	$r_{xj,4}$
$r_{as,1}$	[10, 12, 14, 16, 16, 12, 13, 2]	[8, 10, 12, 14, 14, 10, 11]	[8, 8]	[8, 10, 10]
$r_{xs,1}$	[8, 10, 12, 14, 14, 10, 11]	[6, 8, 10, 12, 12, 8, 9]	[6, 6]	[6, 8, 8]
$r_{xs,3}$	[10, 12, 12, 8, 9]	[8, 10, 10, 6, 7]	[4, 4]	[4, 6, 6]
~~$r_{xs,4}$~~	~~[8, 10]~~	~~[6, 8]~~	~~2~~	~~[2, 4]~~

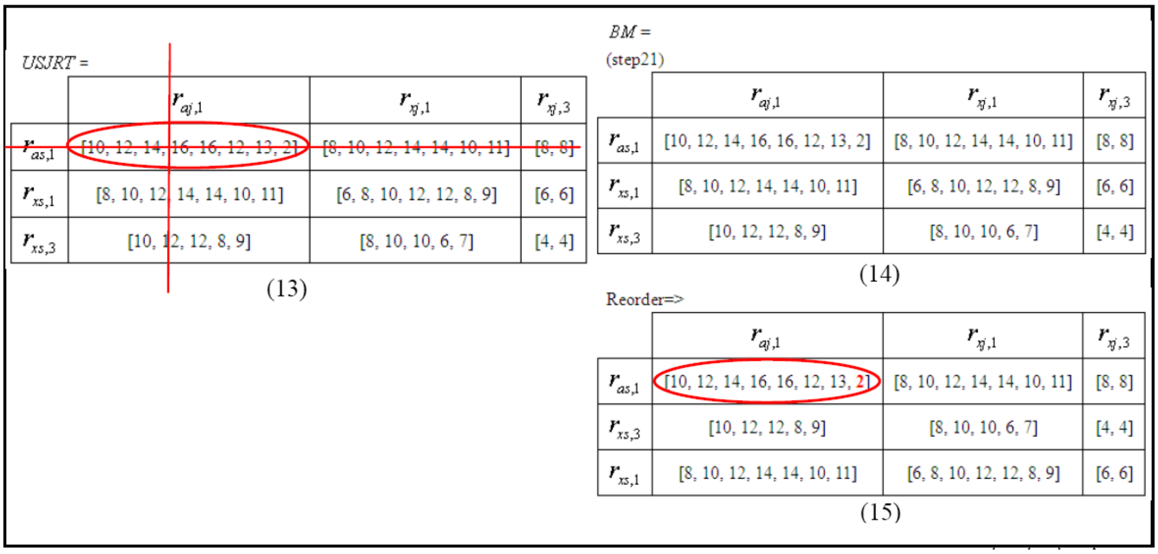
(11)

BM =
(step7)

	$r_{aj,1}$	$r_{xj,1}$	$r_{xj,3}$	$r_{xj,4}$
$r_{xs,4}$	[8, 10]	[6, 8]	2	[2, 4]

(12)

FIGURE 8. Procedure for identifying matched pairs (continued)



USJRT =

	$r_{aj,1}$	$r_{xj,1}$	$r_{xj,3}$
~~$r_{as,1}$~~	~~[10, 12, 14, 16, 16, 12, 13, 2]~~	~~[8, 10, 12, 14, 14, 10, 11]~~	~~[8, 8]~~
$r_{xs,1}$	[8, 10, 12, 14, 14, 10, 11]	[6, 8, 10, 12, 12, 8, 9]	[6, 6]
$r_{xs,3}$	[10, 12, 12, 8, 9]	[8, 10, 10, 6, 7]	[4, 4]

(13)

BM =
(step21)

	$r_{aj,1}$	$r_{xj,1}$	$r_{xj,3}$
$r_{as,1}$	[10, 12, 14, 16, 16, 12, 13, 2]	[8, 10, 12, 14, 14, 10, 11]	[8, 8]
$r_{xs,1}$	[8, 10, 12, 14, 14, 10, 11]	[6, 8, 10, 12, 12, 8, 9]	[6, 6]
$r_{xs,3}$	[10, 12, 12, 8, 9]	[8, 10, 10, 6, 7]	[4, 4]

(14)

Reorder=>

	$r_{aj,1}$	$r_{xj,1}$	$r_{xj,3}$
$r_{as,1}$	[10, 12, 14, 16, 16, 12, 13, 2]	[8, 10, 12, 14, 14, 10, 11]	[8, 8]
$r_{xs,3}$	[10, 12, 12, 8, 9]	[8, 10, 10, 6, 7]	[4, 4]
$r_{xs,1}$	[8, 10, 12, 14, 14, 10, 11]	[6, 8, 10, 12, 12, 8, 9]	[6, 6]

(15)

FIGURE 9. Procedure for identifying matched pairs (continued)

14. *Let* $BM = [a_{ij}^x]$
15. *Reorder the row in BM such that* $\min|a_{1j}^x| \leq \min|a_{2j}^x| \leq \ldots \min|a_{nj}^x|$
16. $\forall\, ||a_{1j}| \neq 1|$, *find* $\min|a_{1j}^x|$ *and let the corresponding* $r_{sp,i}$ *and* $r_{jp,j}$ *form a matched pair* $[r_{sp,i}, r_{jp,j}]$
17. *For* a_{1j}^x *in BM, find the respective* a_{ij} *in USJRT*
18. *Let* $USJRT = USJRT - [r_{sp,i}, r_{jp,j}, a_{ij}]$
19. *Back to Step-13 until BM becomes an empty matrix*

The first step is to identify the rows, such that all enclosed elements have the same number of components (Step 13). These identified rows form a BM (Step 14). Then, the row containing the shortest route will be in the first row (Step 15). The remaining steps are the same as that for the previous step; that is, identify the related element in the first row and select the smallest route value.

The final task is to extract the pattern pairs, such each connected route in a pair may contain a different number of arcs. The algorithm is as follows:

```
20. For $i = 1..n$, find the row $i$ in USJRT, such that $\forall a_{ij}$ in that row, $||a_{ij}| \neq 1|$
21. Let $BM = \left[a^{x}_{ij}\right]$
22. Reorder the row in BM such that $||a_{1j}| \neq 1|_{j=1}^{m} \leq ||a_{2j}| \neq 1|_{j=1}^{m} \leq \dots ||a_{nj}| \neq 1|_{j=1}^{m}$
    and $\min\left|a^{x}_{1j}\right| \leq \min\left|a^{x}_{2j}\right| \leq \dots \min\left|a^{x}_{nj}\right|$
23. $\forall\, ||a_{1j}| \neq 1|$, find $\min\left|a^{x}_{1j}\right|$ and let the corresponding $r_{sp,i}$ and $r_{jp,j}$ form a matched
    pair $[r_{sp,i}, r_{jp,j}]$
24. For $a^{x}_{1j}$ in BM, find the respective $a_{ij}$ in USJRT
25. Let $USJRT = USJRT - [r_{sp,i}, r_{jp,j}, a_{ij}]$
26. Back to Step-13 until BM becomes an empty matrix
```

Step 20 is utilized to identify the row in which the number of routes in each element differs. For example, the identified rows from the USJRT (Figure 9(13)) are put in the BM (Figure 9(14)). The components in these three rows are all multi-value elements: $(||a_{1j}| \neq 1|_{j=1}^{m} = ||a_{2j}| \neq 1|_{j=1}^{m} = ||a_{3j}| \neq 1|_{j=1}^{m} = 3)$. These rows will be reordered, such that the row containing the smallest routing value will be in the leading rows of the BM. In this example, since minimum routing values in these rows are $\min\left|a^{x}_{1j}\right| = |a^{8}_{11}| = 2 \leq \min\left|a^{x}_{3j}\right| = |a^{1}_{33}| = 4 \leq \min\left|a^{x}_{2j}\right| = |a^{1}_{23}| = 6$, rows 2 and 3 in the BM will be switched (Figure 9(15)). The minimum route value occurs in $a_{11}(\min|a^{x}_{11}| = |a^{8}_{11}| = 2)$, and the respective split and join patterns $[r_{as,1}, r_{aj,1}]$ constitute the sixth pattern pair. Since two rows remain in the BM under similar scenarios, the analytical process will be run twice and two other patterns pairs, $[r_{xs,3}, r_{xj,3}]$ and $[r_{xs,1}, r_{xj,1}]$, are identified.

3.2.4. *Identify improper workflow structures.* After constructing an SJRT, the matched pairs as well as paths connected to the split and join pattern in each pair are identified. In the example (Figure 4(2)), the eight matched pairs are $[r_{as,2}, r_{xj,6}]$, $[r_{xs,6}, r_{aj,2}]$, $[r_{xs,2}, r_{xj,2}]$, $[r_{xs,5}, r_{xj,5}]$, $[r_{xs,4}, r_{xj,4}]$, $[r_{as,1}, r_{aj,1}]$, $[r_{xs,3}, r_{xj,3}]$ and $[r_{xs,1}, r_{xj,1}]$. These pairs can then be used to identify improper workflow structures in a model. In this work, three nesting structure analysis types – non-corresponding nesting, improper nesting, and not well-structured analysis – are carried out based on the information extracted from the SRJT.

3.2.4.1. Corresponding nesting analysis. In a structured workflow, each split-parallel element must have a corresponding join-parallel element, and each split-choice element must have a corresponding join-choice element [3]. Since the split patterns and join patterns have been specified in the indexed row and column in the SJRT, a workflow with a non-corresponding nesting structure can be defined as

Definition 3.5. *Non-corresponding Nesting*

If the number of rows is not equal to the number of columns in an SJRT, this workflow structure is a non-corresponding nesting; otherwise, the structure is a corresponding nesting.

Figure 10 shows an example of non-corresponding nesting. This has nine split patterns and six join patterns. Therefore, at least three split patterns cannot find their matched join patterns and, hence generate non-corresponding nesting. However, the PN (Figure 4(1)) is a corresponding nesting because its PBPD is an 8 × 8 matrix.

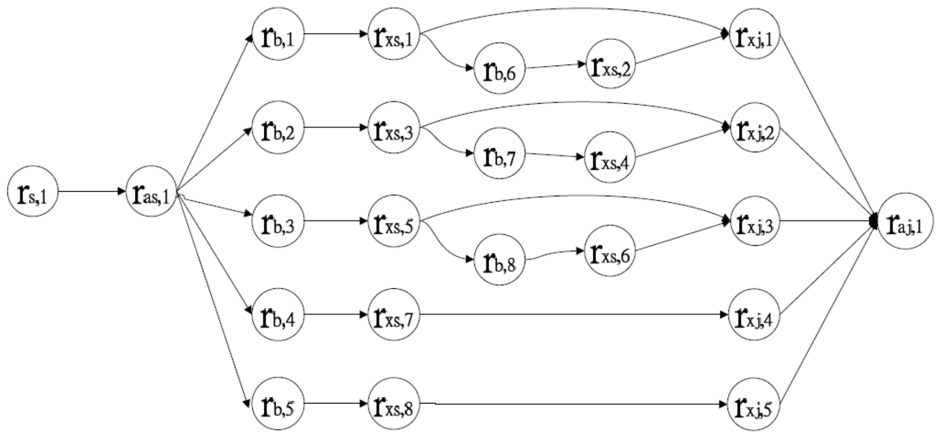


FIGURE 10. Example of non-corresponding nesting process

3.2.4.2. Improper nesting analysis. Among the identified pattern pairs, the paths connecting one pair may interact with a path connecting another pair. This situation, called improper nesting, can be defined as

Definition 3.6. *Improper Nesting*

A workflow is an improper nesting two sets of matched pairs, $[r_{sp,i}, r_{jp,j}]$ *and* $[r_{sp,i'}, r_{jp,j'}]$*, exist, such that one of their connected routes* $[r_{sp,i}, r_{jp,j}] \to a^x_{ij}$ *or* $[r_{sp,i'}, r_{jp,j'}] \to a^y_{i'j'}$ *satisfy the following conditions:*

1. $a^x_{ij} \cap a^y_{i'j'} \neq \varnothing$*, and*
2. $a^x_{ij} \not\subset a^y_{i'j'}$ *or* $a^y_{i'j'} \not\subset a^x_{ij}$

This definition means that for an improper structure, two sets of matched pairs exist, such that their routes overlap ($a^x_{ij} \cap a^y_{i'j'} \neq \varnothing$) but are not mutually included ($a^x_{ij} \not\subset a^y_{i'j'}$ *or* $a^y_{i'j'} \not\subset a^x_{ij}$). If the routes of one matched pair are contained in the routes of another matched pair ($a^x_{ij} \subset a^y_{i'j'}$), these pairs are well structured.

According to this definition, for the two matched pairs, $[r_{xs,5}, r_{xj,5}]$ and $[r_{xs,3}, r_{xj,3}]$, (Figure 4(2)) the two connected paths, $[r_{xs,5}, r_{xj,5}] \to a^1_{55}$ and $[r_{xs,3}, r_{xj,3}] \to a^1_{33}$ can be found, such that $a^1_{55} = [r_{xs,5} \to r_{b,11}, r_{b,11} \to r_{xj,3}, r_{xj,3} \to r_{b,13}, r_{b,13} \to r_{xj,4}, r_{xj,4} \to r_{b,15}, r_{b,15} \to r_{xj,5}]$ and $a^1_{33} = \{[r_{xs,3} \to r_{b,7}, r_{b,7} \to r_{xs,5}, r_{xs,5} \to r_{b,11}, r_{b,11} \to r_{xj,3}]$ overlap during the arcs $r_{xs,5} \to r_{b,11}$ and $r_{b,11} \to r_{xj,3}$. Therefore, this model has improper nesting.

3.2.4.3. Not well-structured analysis. For a structured PN model, a one-to-one correspondence relationship (i.e., correspond to the same type) between a split element and a join element must be satisfied. That is, each AND-Split has a corresponding AND-Join and each XOR-Split has a corresponding XOR-Join. A workflow is said to be not well structured if an AND-Split is matched with an XOR-Join, or an XOR-Split is matched with an AND-Join. The situations of *deadlocks* and *multiple instances of the same activity* occur when the corresponding pattern types are not matched [3,8]. Since the pattern types for each matched pair can be identified using the proposed algorithm, the not well-structured workflow model can be defined as

Definition 3.7. *Not well structured*

A workflow is not well structured if a matched pair $[r_{sp,i}, r_{jp,j}]$ *exists, such that* $sp = as$ *and* $jp = xj$*, or* $sp = xs$ *and* $jp = aj$*.*

For instance, in the PBPD (Figure 4(2)), the patterns in first and second matched pairs, ($[r_{as,2}, r_{xj,6}]$ and $[r_{xs,6}, r_{aj,2}]$), do not belong to the same type (AND-Split *vs.* XOR-Join,

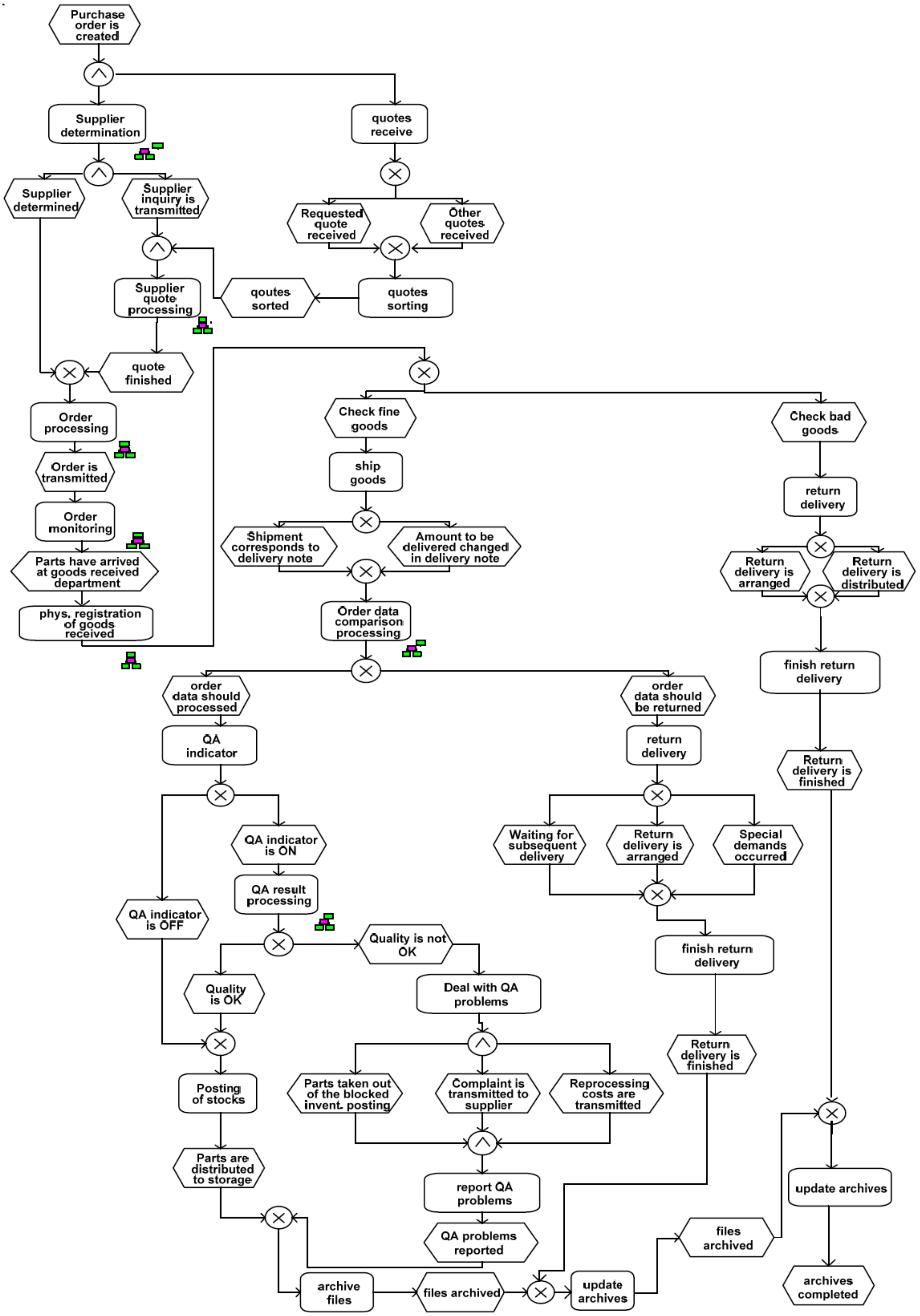


Figure 11. The modified reference model for acquisition handling

and XOR-Split *vs.* AND-Join); therefore, deadlock and multiple instances occur in this model.

From this analysis, the example PN model (Figure 4(1)) is a not well structured, improper nesting but corresponding nesting model.

The complexity of the proposed algorithm is $O(l!)$, where l is the number of join patterns, and the most complicated part occurs in "identification of matched pairs" stage. That is, the routes among each pattern must be calculated and searched while Split-join routing table is formed. As the number of the patterns grows, the number of steps required to complete the search increases dramatically and hence causes heavy calculation.

4. Implementation.

A workflow structure analysis tool, the Pattern Identifier and Process Structure Analyzer (PIPSA), was developed to implement the proposed concepts. The XML file of the empirical PN model is the input for the PIPSA. This tool has three main components – (a) a Pattern Identifier component that discovers workflow patterns, (b) Process Structure Construction component that uses identified workflow patterns to construct a PBPD and an SJRT, and (c) Process Structure Analysis module that identifies pattern pairs from the SJRT and analyzes the structure of the workflow model.

4.1. Empirical test.

An "acquisition handling" EPC model from the ARIS "Mechanical Engineering Reference Model" library was selected as an empirical model. This model was developed based on ARIS industrial experience. Details of the ARIS business reference model can be found in the study by Scheer [22].

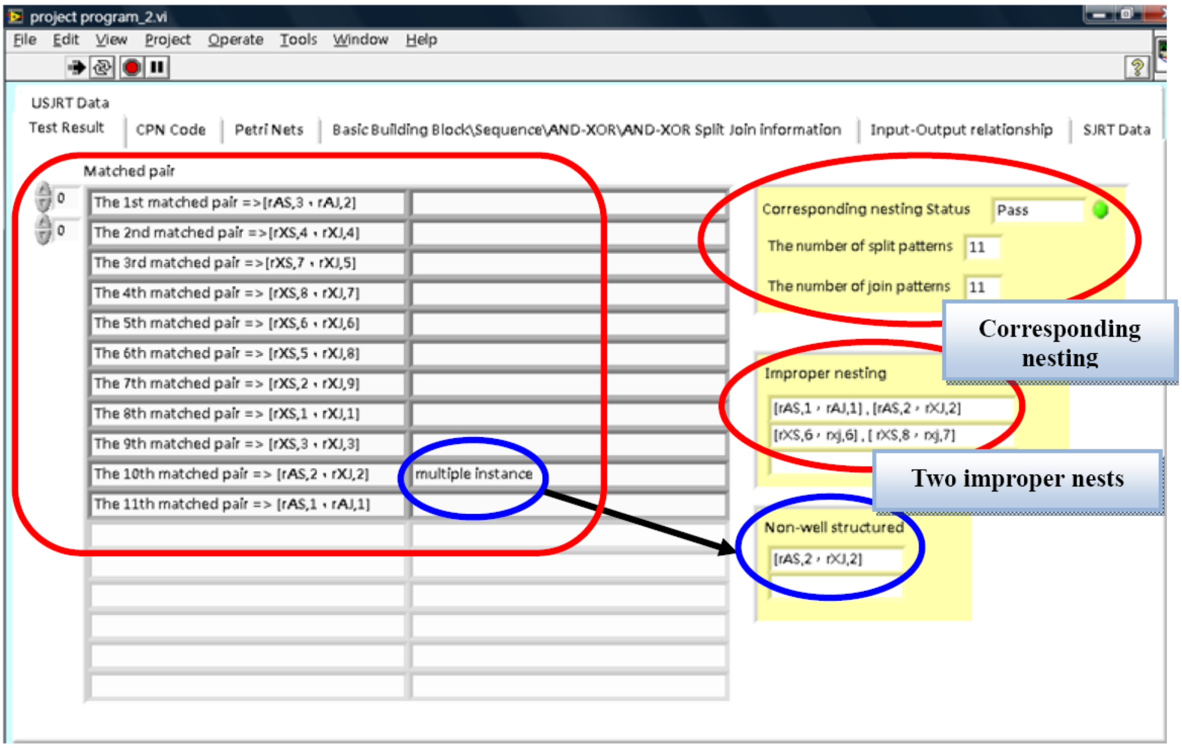
An initial analysis the model indicates that several input and output events representing various documents are required in this process. This violates the requirements of the workflow net, which only allows for one initial node and one final node. Human adjustment is carried out to group input events through XOR-Split, and output events through XOR-Join. This modification does affect the structure of the original model. Figure 11 shows the modified "acquisition handling" EPC model. This model is then transformed into a PN model using the EPC/PN mapping approach developed by van der Aalst et al. [23].

This model constructed using the Colored Petri Net (CPN) tool is imported into PIPSAT using CPNXML. Figure 12 shows the analytical results.

4.2. Empirical model analysis and re-design.

Based on the identified pattern pairs, two improper nesting structures existed. After analyzing the PBPD, overlapping connecting arcs exist in these pairs. In the real case, one sub-process has merged with another before the sub-process has finished. When compared with the corresponding EPC model, the "supplier determination" sub-process has mixed with the "quote received" in one improper nesting structure. Therefore, the PN model is transformed into a structured model using a mapping approach [3]. Figure 13 shows the mapped EPC model. The "quote received" sub-process has connected after finishing the "supplier determination" sub-process but before execution of the "order processing" function. The logic of the original reference model remains in the modified EPC model.

In addition to the identified improper nesting structures, a not well-structured pair is found. Since the patterns are AND-Split/XOR-Join pairs, multiple instances may occur during the acquisition handling process. By further analyzing the PN and respective EPC model, this situation takes place between "supplier determination" and "order processing" functions. Via a human examination of the EPC model, this may be a fault; that is, the "order process" should be carried out after the "supplier determined" and "quote finished" events are completed. Therefore, the XOR-Join should be replaced by an AND-Join.



Matched pair

0

0

The 1st matched pair =>[rAS,3 , rAJ,2]

The 2nd matched pair =>[rXS,4 , rXJ,4]

The 3rd matched pair =>[rXS,7 , rXJ,5]

The 4th matched pair => [rXS,8 , rXJ,7]

The 5th matched pair => [rXS,6 , rXJ,6]

The 6th matched pair => [rXS,5 , rXJ,8]

The 7th matched pair => [rXS,2 , rXJ,9]

The 8th matched pair => [rXS,1 , rXJ,1]

The 9th matched pair => [rXS,3 , rXJ,3]

The 10th matched pair => [rAS,2 , rXJ,2] multiple instance

The 11th matched pair => [rAS,1 , rAJ,1]

Corresponding nesting Status Pass

The number of split patterns 11

The number of join patterns 11

Improper nesting

[rAS,1 , rAJ,1] , [rAS,2 , rXJ,2]

[rXS,6 , rxj,6] , [rXS,8 , rxj,7]

Non-well structured

[rAS,2 , rXJ,2]

FIGURE 12. Identified three kinds of improper workflow structures

TABLE 1. The comparison of pattern reduction works

Reference	Logic	Routing interpretation	Taxonomy of unstructured workflows			
			Non-corresponding nesting	Deadlocks	Multiple instances	Improper nesting
Sadiq et al. [14]	√			√	√	
Aalst et al. [15]	√			√	√	
Lee et al. [16]	√					
Sloan et al. [17]	√					
Xia [18]	√					
Proposed approach	√	√	√	√	√	√

5. **Conclusions.** A BPR project, such as implementing an ERP system, typically requires a reference model that functions as a TO-BE model. Although a reference model can be obtained from various resources, an approach is required to analyze the model structure and re-design it, such that the modified model will have enhanced feasibility.

A PN has been widely used in business process modeling. However, due to the limited types of constructs that can be employed in this methodology, workflow models with complex structures are not rare. This has increased the difficulty of tasks in process structure analysis. Therefore, to reduce process model complexity and assist in structure analysis, this work developed a pattern-based workflow structural analysis approach.

The works about pattern reduction and structural analysis are compared in Table 1. Most of the works aim to reduce graphs or PNs and retain logic in original PNs. In addition, some works deal with the structural analysis issues such as deadlocks and multiple instances. However, reduced models may not maintain the routing structure of the original process models. They may lose the dependencies among activities (the causal relationships among transitions and places) during reducing stage, and hence might be difficult to clarify the consequence of the process path.

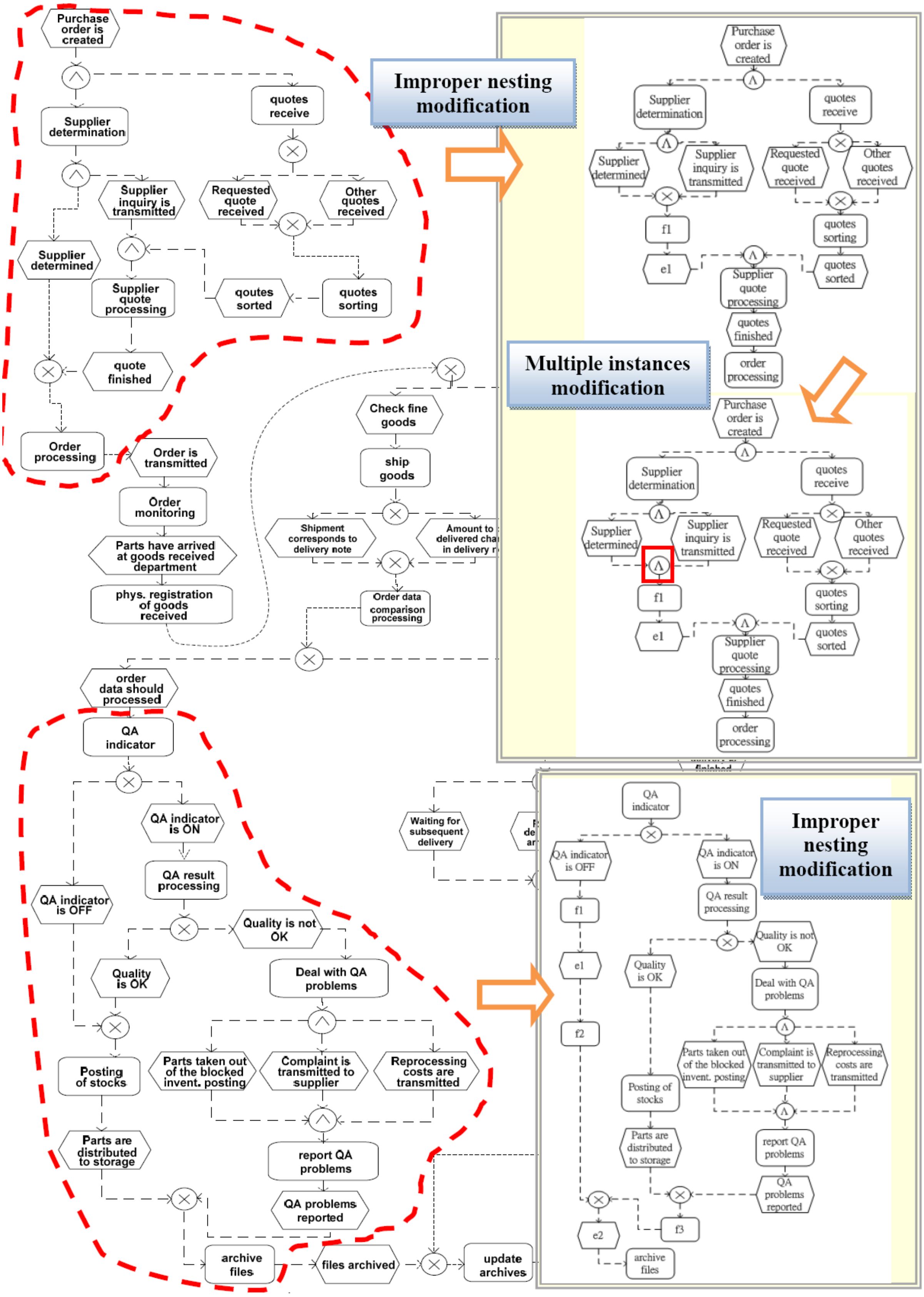


Figure 13. Mapped EPC reference model after modification

The contributions of this work are as follows.

1. The proposed PBPD can simplify a complex PN model and maintain the routing structure of the original PN model during analysis.

2. A novel pattern-pair identification approach is proposed based on the SJRT extracted from the PBPD.

3. A workflow structural analysis approach based on recognized pattern-pairs has been used to identify several structural problems in a workflow model.

REFERENCES

[1] J. S. Bae, S. C. Jeong, Y. Seo, Y. Kim and S. H. Kang, Integration of workflow management and simulation, *Computers and Industrial Engineering*, vol.37, pp.203-206, 1999.

[2] D. W. Lee, H. Shin and B. K. Choi, Mediator approach to direct workflow simulation, *Simulation Modelling Practice and Theory*, vol.18, no.5, pp.650-662, 2010.

[3] R. Liu and A. Kumar, An analysis and taxonomy of unstructured workflows, *Proc. of the International Conference on Business Process Management*, vol.3649, pp.268-284, 2005.

[4] W. M. P. van der Aalst, A. H. M. ter Hofstede, B. Kiepuszewski and A. P. Barros, Advanced workflow patterns, *The 7th International Conference on Cooperative Information Systems (CoopIS 2000)*, vol.1901, pp.18-29, 2000.

[5] W. M. P. van der Aalst, A. H. M. ter Hofstede, B. Kiepuszewski and A. P. Barros, Workflow patterns, *Distributed and Parallel Databases*, vol.14, no.1, pp.5-51, 2003.

[6] W. Gaaloul, K. Baïna and C. Godart, Towards mining structural workflow patterns, *DEXA*, vol.3588, pp.24-33, 2005.

[7] G. Zhou, Y. He and P. Yu, Modelling workflow patterns based on P/T nets, *International Journal of Innovative Computing, Information and Control*, vol.1, no.4, pp.673-684, 2005.

[8] B. Kiepuszewski, A. H. M. ter Hofstede and C. Bussler, On structured workflow modeling, *Proc. of CaiSE, LNCS*, vol.1797, 2000.

[9] I. Choi, C. Park and C. Lee, A transactional workflow model for engineering/manufacturing processes, *International Journal of Computer Integrated Manufacturing*, vol.15, no.2, pp.178-192, 2002.

[10] P. A. Buhler and J. M. Vidal, Towards adaptive workflow enactment using multiagent systems, *Information Technology and Management Journal*, vol.6, no.1, pp.61-87, 2005.

[11] P. Sun and C. Jiang, Analysis of workflow dynamic changes based on Petri net, *Information and Software Technology*, vol.51, no.2, pp.284-292, 2009.

[12] G. A. Chaparro-Baquero, N. G. Santiago, W. Rivera and J. F. Vega-Riveros, Measuring quantitative dependability attributes in digital publishing using Petri net workflow modeling, *Proc. of the 2nd IEEE International Symposium on Dependable, Autonomic and Secure Computing*, pp.119-128, 2006.

[13] R. Maeno, M. Konishi and J. Imai, Petri net solver for semiconductor manufacturing plan, *International Journal of Innovative Computing, Information and Control*, vol.4, no.12, pp.3193-3206, 2008.

[14] W. Sadiq and M. E. Orlowska, Analyzing process models using graph reduction techniques, *Information Systems*, vol.25, no.2, pp.117-134, 2000.

[15] W. M. P. van der Aalst, A. Hirnschall and H. M. W. Verbeek, An alternative way to analyze workflow graphs, *Proc. of the 14th Int. Conf. on Advanced Information Systems Eng., LNCS*, vol.2348, pp.534-552, 2002.

[16] K. H. Lee, J. Favrel and P. Baptiste, Generalized Petri net reduction method, *IEEE Trans. Syst. Man Cybern*, vol.17, pp.297-303, 1987.

[17] R. H. Sloan and U. Buy, Reduction rules for time Petri nets, *Acta Informatica*, vol.33, no.7, pp.687-706, 1996.

[18] C. Xia, Reduction rules for Petri net based representation for embedded systems, *ATPN's Workshop: Protocols Engineering Based on Petri Nets and Modeling of Concurrent Systems*, pp.16-33, 2008.

[19] H. Lee and H. W. Suh, Workflow structuring and reengineering method for design process, *Computers and Industrial Engineering*, vol.51, no.4, pp.698-714, 2006.

[20] B. F. van Dongen, M. H. Jansen-Vullers, H. M. W. Verbeek and W. M. P. van der Aalst, Verification of the SAP reference models using EPC reduction, state-space analysis, and invariants, *Computers in Industry*, vol.58, no.6, pp.578-601, 2007.

[21] T. Murata, Petri nets: Properties, analysis and applications, *Proc. of the IEEE*, vol.77, no.4, pp.541-580, 1989.

[22] A. W. Scheer, *Business Process Engineering: Reference Models for Industrial Enterprises*, Springer-Verlag, 1994.

[23] W. M. P. van der Aalst, Formalization and verification of event-driven process chains, *Information and Software Technology*, vol.41, no.10, pp.639-650, 1999.