International Journal of Grid and Distributed Computing
Vol. 9, No. 4 (2016), pp.121-134
http://dx.doi.org/10.14257/ijgdc.2016.9.4.12

Business Process Adaptation Based on Process Variants*

Hong Xia Li[1, 2], Yu Yue Du[1]

1. College of Information Science and Engineering,
Shandong University of Science and Technology, Qingdao 266590, China
2. College of Science and Information, Qingdao Agricultural University
Qingdao, 266109, China
lovelymelody@163.com, yydu001@163.com

Abstract

An extensional method for existing business processes is proposed to adapt new environments based on process variants in this paper. The core processes describing maximal commonalities of process variants are introduced according to inheritance relation. Then, extensional business processes are defined based on the core processes. The algorithms of the core processes and extensional business are given and proved. And the soundness of two kinds of processes is justified. An example shows that extensional business processes can adapt to the new environment better than the original process.

***Keywords**: commonality; inheritance relation; core process; extensional business process*

1. Introduction

Business Process Management (BPM) combines the knowledge from information technology and the one from management sciences, and is applied to operational business processes [1]. Many enterprises have many business processes, and the same business process may exist in the form of multiple variants [2]. New business processes can be constructed easily from process variants, but variant management is difficult.

Many methods are proposed to regulate and reuse similar business processes, *i.e.* process variants, such as a reference process model [3] and a configurable process model [4-5]. A reference process model tries to capture recurrent processes to gain a standard reference model repository from the process variants, and a specific process model can be realized manually for individualization. A configurable process model represents multiple business process model variants in an integrated manner. It can be built by recognizing [6-7] and merging [8] similar business processes. A specific process model is decided at the build time or run time [9].

The above models are based on the hypothesis that an individual business process is an alternative of the integrated business process model. In fact, it is difficult to get a specific process directly from existing process variants, and it is often necessary to adapt a business process to specific environment like specific law or company culture [9]. In this paper, core processes are proposed to denote maximal commonalities of process variants, and a business process can be extended based on core processes to meet specific environment requirement.

In term of modeling language, there exist many different notations for process models, such as labeled transition systems, Event-driven Process Chains (EPC), Business Process Modeling Notation (BPMN), Petri nets, and workflow nets (WF-nets). Petri nets have been extensively applied to the formal verification of business process models [10-14].

[*] Corresponding author: Yu Yue Du. Email: yydu001@163.com

ISSN: 2005-4262 IJGDC

WF-nets [11] allow for a clear definition with initial and final states to describe the process start and end. Therefore, we choose WF-nets to model the process in this paper.

The rest of the paper is structured as follows. Section 2 introduces preliminaries. Section 3 discusses the concepts and algorithms of the core process (CP) and extensional business process (EBP). Then, the algorithms and the soundness of these processes are proved. In section 4, declare tax processes are introduced to illustrate the extensional method of business processes. The related works and conclusions are shown in Sections 5 and 6 respectively.

2. Preliminaries

Some concepts are introduced in this section. The definition of labeled P/T–nets [15] is reviewed firstly. Then behavioral equivalence and inheritance relation are presented.

Definition 1 (*Labeled P/T-net*) A labeled P/T-net is a class of Petri nets, described as a tuple $LN= (P, T, F, L)$, where

(1) P is a finite set of *places*;
(2) T is a finite set of *transitions*, such that $P\cup T\neq\emptyset$, $P\cap T=\emptyset$;
(3) $F \subseteq (P\times T) \cup (T\times P)$ is a set of directed arcs of P and T, called the *flow relation*;
(4) $dom(F)\cup cod(F)=P\cup T$ where
(5) $dom(F)=\{x\in P\cup T | \exists y\in P\cup T: (x, y)\in F\}$
(6) $cod(F)=\{y\in P\cup T | \exists x\in P\cup T: (x, y)\in F\}$ and
(7) L: $T\rightarrow A\cup\tau$ is a *labeling function*, and A is a set of activities and τ denotes an invisible activity.

Definition 2 *(Pre-set and Post-set)* Let LN be a labeled P/T-net, $x\in P\cup T$ be an element of LN, then

$^{\bullet}x=\{y | y\in P\cup T\wedge(y, x) \in F\}$ is called the pre-set of x;
$x^{\bullet}=\{y | y\in P\cup T\wedge(x, y) \in F\}$ is called the post-set of x.

A class of Petri nets, namely workflow nets, is specifically designed for business process modelling. Each workflow net has a single starting point and an ending point, which represents the creation and completion of a business process. Each process instance progresses independently through a series of activities until completion. The definition of labeled WF-net is described in the following, and a silent transition being not observed is labelled by the symbol τ.

Definition 3 *(Labeled WF-net)* Let $LN= (P, T, F, L)$ be a labeled P/T-net. And t' is not in $P\cup T$ and labelled by τ.Then N is a labeled Workflow net (labeled WF-net) iff:

(1) Process creation: it has an only source place i and $^{\bullet}i=\emptyset$;
(2) Process completion: it has an only sink place o and $o^{\bullet}=\emptyset$; and
(3) Connectedness: $N'= (P, T \cup\{t'\}, F\cup\{(o, t'), (t', i)\}, L\cup\{(t',\tau)\})$ is strongly connected.

Definition 4 *(Firing rule)* Let $N= (P, T, F, L)$ be a labeled WF-net with source place i and sink place o:

(1) M: $P\rightarrow\{0,1, 2, \ldots\}$ is a *marking* of N;
(2) M_i is the initial marking of N with one token in i, *i.e.* $M_i= [i]$;
(3) M_o is the final marking of N with one token in o, *i.e.* $M_o= [o]$;
(4) For all p, $p\in{}^{\bullet}t$, if $M (p) \geq 1$, then t is called *enabled* under M, denoted as $M [t>$. If $M [t>$ and t is fired, then a new marking M' is generated, denoted by $M [t> M'$, where

$$M'(p)=\begin{cases} M(p)+1 & \text{if } p\in t^{\bullet}-{}^{\bullet}t \\ M(p)-1 & \text{if } p\in {}^{\bullet}t-t^{\bullet} \\ M(p) & \text{else} \end{cases}$$

The set of all labeled WF-nets is denoted $\Phi(N)$.

To distinguish the descriptive power of different labeled WF-nets, inheritance relation and behavioral equivalence are introduced. Branching bisimilarity [16] is chosen as the standard equivalence relation due to not observed transitions. Branching bisimilarity is defined as follows.

Definition 5 ***(branching bisimilarity)*** Let $N= (P, T, F, L)$ be a labeled WF-net, and a binary relation $R \subseteq(\Phi(N)\times \Phi(N))$ is called a *branching bisimulation* iff, for any p, p', q, $q'\in \Phi(N)$ and $a\in A\cup\{\tau\}$:

(1) $pRq\wedge p[a>p'\Rightarrow\exists q':(q[a>q')\wedge(p'Rq')$;
(2) $pRq\wedge q[a>q'\Rightarrow\exists p':(p[a>p')\wedge(p'Rq')$; and
(3) $pRq\Rightarrow p$ can properly complete iff q can properly complete.

Two labeled WF-nets are called *branching bisimilar*, denoted $p\sim q$, if and only if there is a *branching bisimulation R* such that pRq. Branching bisimilarity is an equivalence relation on $\Phi(N)$, *i.e.* ~ is reflexive, symmetric, and transitive [16].

The behavioral equivalence between labeled WF-nets based on branching bisimilarity can be defined as follows.

Definition 6 ***(behavioral equivalence)*** Let N_1 and N_2 be labeled WF-nets, then N_1 and N_2 satisfy behavioral equivalence iff $(N_1, [i_1]) \sim (N_2, [i_2])$.

Hiding and blocking operation are introduced to define inheritance relation. Hiding a transition refers to skipping its execution while it is fired, without affecting the rest of the process. Blocking a transition implies to inhibit it in the process model. Blocked transitions cannot forward processes and all the subsequent transitions will never be executed unless they can be enabled via other paths [5].

Definition 7 ***(hiding operation)*** Let $N= (P, T, F, L_0)$ be a labeled WF-net, for any $I\subseteq A$ $(L_0{:}T\rightarrow A\cup\tau)$, hiding operation τ_I can be described as follows.

$\tau_I(N) = (P, T, F, L_1)$, where
(1) If $L_0(t)\in I$, then $L_1(t)=\tau$; and
(2) If $L_0(t)\notin I$, then $L_1(t) = L_0(t)$.

Definition 8 ***(blocking operation)*** Let $N= (P_0, T_0, F_0, L_0)$ be a labeled WF-net, for any $H\subseteq A_0$ $(L_0{:}T_0\rightarrow A_0\cup\tau)$, blocking operation ∂_H can be described as follows.

$\partial_H(N)=(P_1, T_1, F_1, L_1)$, where
(1)$T_1= \{t|\ t\in T_0\wedge L_0(t)\notin H\}$
(2) $P_1\subseteq P_0$;
(2)$F_1= F_0\cap((P_1\times T_1)\cup(T_1\times P_1))$; and
(3)$L_1= L_0\cap(T_1\times A)$.

Inheritance relation is defined based on the hiding operation, blocking operation and behavioral equivalence. A labeled WF-net inherits from another labeled WF-net if and only if they satisfy behavioral equivalence by blocking and/or hiding its one or more transitions.

Definition 9 ***(inheritance relation)*** Let N_1 and N_2 be labeled WF-nets. If $\exists I$ and H, such that $(\tau_I\circ\partial_H(N_1),[i])\sim(N_2,[i])$, where I, $H\subseteq A_1$ and $I\cap H=\varnothing$, then N_1 and N_2 are called satisfy *inheritance relation*. Meanwhile, N_1 is a *subclass* of N_2, denoted as $N_1\leq N_2$, and N_2 is a *superclass* of N_1.

3. Adaptation of Business Processes based on the Core Process

Sometimes, it is necessary to adapt business processes to a new environment, for example, the enterprise moves to another country. The existing process variants in the new country are helpful to the adaption of business processes. Firstly, a CP is proposed to describe the maximal commonalities of business process variants in the new environment. Then, an EBP is defined to describe the process adaption based on the CP. The EBP keeps the characteristic of original business process and adapt to the new environment well. The concepts and algorithms of the CP and EBP are given, and the soundness of algorithms and processes are verified in this section.

3.1 Construct CP

A CP, which describes the maximal commonalities of business process variants, should capture maximum activities and their relations of process variants. And the CP can be defined as follows.

Definition 10 ***(Core Process, CP)*** Let N_0= (P_0, T_0, F_0, L_0) be a labeled WF-nets with only sequential structure. Let N_i= (P_i, T_i, F_i, L_i) (1≤i≤n) be labeled WF-nets of process variants. Then a labeled WF-net N= (P, T, F, L) is a CP of N_i iff:

(1) $N_i \leq N$ (1≤ i ≤n), and

(2) If $\exists N_0$ such that $N_i \leq N_0$(1≤ i ≤n), then $N \leq N_0$.

Note that a core process is a superclass of process variants. Furthermore, for any a superclass of process variants, a core process is a subclass of the superclass, *i.e.* a core process embodies the maximal commonalities of process variants.

The set of CPs of process variants is denoted *CPset.*

The execution of a process instance should satisfy proper creation and proper completion without deadlock. A labeled WF-net fulfilling these requirements is *sound*. Formally:

Definition 11 ***(sound)*** Let N= (P, T, F, L) be a labeled WF-net. N is *sound* iff:

(1) Proper completion: for any marking $M \in [N, [i]>$, $o \in M$ implies M=[o];

(2) Option to complete: for any marking $M \in [N, [i]>$, $[o] \in [N, M>$; and

(3) No dead transitions: (N, [i]) contains no dead transitions.

A *trace* of a labeled WF-net is put forth to describe a process activity sequence. And CPset of process variants can be acquired according to the traces.

Definition 12 ***(trace)*** Let N be a labeled WF-net, and $\sigma=[i]t_1M_1t_2M_2t_3...M_{m-1}t_m[o]$ be a finite occurrence sequence. And the corresponding activity sequence of the occurrence sequence $ŧ_N=L(t_1)L(t_2)L(t_3)\ldots L(t_m)$ is a *trace* of N. And $L(t_i)$ (1≤i≤m) is the small component unit of $ŧ_N$, $L(t_i) \in ŧ_N$ (1≤i≤m).

The trace set of a labeled WF-net N is denoted $Ŧ_N$.

Let N_1 and N_2 be labeled WF-nets, then each trace of N_1 and N_2 can be expressed as a string of $L(t)$. Thus, many string functions can be applied in traces, such as *string concatenation* (*strcat*), *substring, strlength* (*strlen*), *longest common substring* (*lcs*). The result of one string *strcat* τ is the string itself. For the trace sets $Ŧ_{N_1}$ of N_1 and $Ŧ_{N_2}$ of N_2, the longest common substring set of N_1 and N_2 is denoted $LCS(Ŧ_{N_1}, Ŧ_{N_2})$.

Let N_1, N_2 be process variants of a common process. The CPset of N_1 and N_2 can be constructed in the following.

```
Algorithm 1 Construct CPset
Input: Labeled WF-nets $N_1$= ($P_1$, $T_1$, $F_1$, $L_1$) and $N_2$= ($P_2$, $T_2$, $F_2$, $L_2$)
Output: $\Phi$
```

```
Begin
S1: Map $i_1$, $i_2$ to $i$ and map $o_1$, $o_2$ to $o$, where $i_1, o_1 \in P_1$ and $i_2, o_2 \in P_2$.
S2: To $N_1$ and $N_2$, find $LCS(Ŧ_{N1}, Ŧ_{N2})$.
S3: If $LCS(Ŧ_{N1}, Ŧ_{N2})$ is empty, then
    {$N=(P, T, F, L)$, where
        {$P= \{i,o\}$, $T=\{t\}$, $F=\{(i,t),(t,o)\}$, $L=\{(t, \tau)\}$, $^\bullet i=\varnothing$, $o^\bullet=\varnothing$},
          $\Phi=\{N\}$, go to S7.}
S4: If $\exists lcs_1$ and $lcs_2 \in LCS(Ŧ_{N1}, Ŧ_{N2})$, such that $lcs_2$ is a substring of $lcs_1$, then
          $LCS(Ŧ_{N1}, Ŧ_{N2}) =LCS(Ŧ_{N1}, Ŧ_{N2}) -\{lcs_2\}$.
S5: For each $lcs$ in $LCS(Ŧ_{N1}, Ŧ_{N2})$ do
    {If $len(lcs)=1$ and $lcs==L(t_1)$ then
         {$N=(P, T, F, L)$, where
            {$P=\{i,o\}$, $T=\{t_1\}$, $F=\{(i,t_1),(t_1,o)\}$, $L=\{(t_1, L(t_1))\}$, $^\bullet i=\varnothing$, $o^\bullet=\varnothing$}.
            $\Phi= \Phi\cup N$}
    Else let $lcs== lcs_1$ strcat $L(t)$ , and $lcs_1$ corresponds to $N'= (P', T', F', L')$
      {For each $t' \in {^\bullet o}$
      {$lcs$ corresponds to $N=(P, T, F, L)$ where
      {$P=P'\cup\{p\}$, $T=T'\cup\{t\}$, $F=F'\cup\{(t', p), (p,t), (t,o)\}-\{(t', o)\}$, $L=L'\cup\{(t, L(t))\}$}}
      $\Phi= \Phi\cup N$
        EndFor}
        EndIF}
      EndFor
S6: $LCS(Ŧ_{N1}, Ŧ_{N2})= LCS(Ŧ_{N1}, Ŧ_{N2})-\{lcs\}$
S7: Output $\Phi$
End
```

Figure 1. The CPset Construction Algorithm

In Algorithm 1, $LCS(Ŧ_N1, Ŧ_N2)$ is used to construct Φ. So, each element in Φ captures the maximal commonalities of N_1 and N_2. Following conclusion proves that what we got by Algorithm 1 is the CPset of N_1 and N_2.

Theorem 1 Let $N_1= (P_1, T_1, F_1, L_1)$ and $N_2= (P_2, T_2, F_2, L_2)$ be labeled WF-nets of a business process. $N_0= (P_0, T_0, F_0, L_0)$ is a labeled WF-net with only sequential structure. Let $N= (P, T, F, L)$ be an element of Φ by Algorithm 1, then N is a CP of N_1 and N_2.

Proof. (1) Since N just has the sequential structure (Algorithm 1), *i.e.* N has an only occurrence sequence $\sigma=[i]t_1M_1t_2M_2t_3\ldots M_{m-1}t_m[o]$, obviously, N has an only source place i and sink place o and $^\bullet i=\varnothing$, $o^\bullet=\varnothing$. Meanwhile, Let $N'= (P, T\cup\{t'\}, F\cup\{(o,t'), (t',i)\}, L\cup\{(t',\tau)\})$ where t' is not in $P\cup T$, then N' is strongly connected. So, N is a labeled WF-net.

(2) In N, for $\forall t\in T$ such that $L(t)\in A$, there must exist a trace $ŧ_{N1}$ of N_1 and a trace $ŧ_{N2}$ of N_2, such that $L(t)\in ŧ_{N1}$ and $L(t) \in ŧ_{N2}$.

Let $Ŧ_{N1}$ be the trace set of N_1. For N_1, let $I=\{L_1(t)|L_1(t)\in ŧ_{N1}\wedge L_1(t)\notin A\}$, $lcs_1\in Ŧ_{N1}-\{ŧ_{N1}\}$ and $H=\{L_1(t)| L_1(t)\in lcs_1\wedge L_1(t)\notin A\}$, then $(\tau_I \circ\partial_H (N_1),[i])\sim(N,[i])$, *i.e.* $N_1 \leq N$.

$N_2 \leq N$ can be proved similarly.

(3) If $\exists N'= (P', T', F', L')$ such that $N_1 \leq N'$, $N_2 \leq N'$. For $\forall t'\in T'$ such that $L(t')\in A'$, there are $L(t')\in A_1$ and $L(t')\in A_2$, *i.e.* $\exists lcs_2\in LCS(Ŧ_{N1}, Ŧ_{N2})$ such that $L(t')\in lcs_2$. Let $I=\{L(t)|L(t)\in lcs_2 \wedge L(t)\notin A'\}$, $(\tau_I(N),[i])\sim(N',[i])$, *i.e.* $N\leq N'$.

Therefore, N is a CP of N_1 and N_2 (Definition 10). Because N is any an element of Φ, Φ is the CPset of N_1 and N_2.

According to Algorithm 1, there may be many core processes for two labeled WF-nets, and there is only sequential structure in each CP. Meanwhile, Algorithm 1 can be easily extended to multiple labeled WF-nets. If core processes of multiple process variants are gained, then all of them are sound. The core processes derived from Algorithm 1 can preserve soundness in the following theorem.

Theorem 2 Let $N_i = (P_i, T_i, F_i, L_i)$ $(1 \leq i \leq m)$ be labeled WF-nets of a business process and $N= (P, T, F, L)$ be a CP of N_i. Then N is sound.

Proof. Since $N= (P, T, F, L)$ has an only occurrence sequence $\sigma=[i]t_1M_1t_2M_2t_3...M_{m-1}t_m[o]$, an only token moves from i towards o via each $t \in T$.

(1)Proper completion: when o is marked, all other places are empty, *i.e.* $o \in M$ implies $M=[o]$.

(2)Option to complete: for any $M \in [N,[i]>$, the only token move toward to o, *i.e.* $[o] \in [N,M>$.

(3)No dead transitions: each t is on the only path from i to o, *i.e.* $M[t>$ and t is not dead.

Therefore, N is sound.

In Figure 2, $Ŧ_{N1}=\{ABC\}$, $Ŧ_{N2}=\{ADC\}$ and $LCS(Ŧ_{N1}, Ŧ_{N2})=\{AC\}$. Since $(\tau_{\{B\}}(N_1),[i])\sim(CP_1,[i])$ and $(\tau_{\{D\}}(N_2),[i])\sim(CP_1,[i])$, then $N_1 \leq CP_1$ and $N_2 \leq CP_1$ (Definition 9). In Figure 3, $Ŧ_{N1}=\{ABC\}$, $Ŧ_{N2}=\{ABC, ADC\}$ and $LCS(Ŧ_{N1}, Ŧ_{N3})=\{ABC\}$. Because of $(N_1, [i])\sim(CP_2,[i])$ and $(\partial_{\{D\}}(N_3), [i])\sim(CP_2,[i])$, $N_1 \leq CP_2$ and $N_3 \leq CP_2$ (Definition 9). CP_1 is the CP of N_1, N_2, and CP_2 is the CP of N_1 and N_3 (Theorem 1). Furthermore, from Theorem 2, CP_1 and CP_2 are sound.

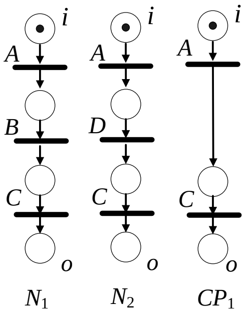


Figure 2. The CP of N_1 and N_2

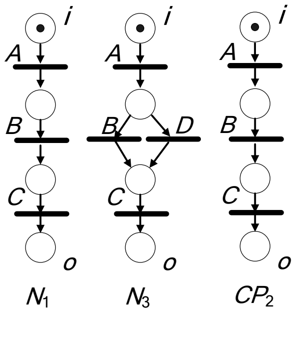


Figure 3. The CP of N1 and N3

3.2 Business Process Adaptation

After core processes describing the commonalities of process variants are found, a business process can be adjusted to adapt specific environment like specific law or company culture based on these core processes. The business process can be extended based on core processes, and extensional business processes are obtained.

Definition 13 ***(Extensional business process, EBP)*** Let N_1= (P_1, T_1, F_1, L_1) be a labeled WF-net of a business process and N_2= (P_2, T_2, F_2, L_2) be a CP of process variants. Let N'= (P', T', F', L') be a labeled WF-net such that $N'\leq N_1$ and $N'\leq N_2$. Then a labeled WF-net EBP= (P, T, F, L) is an EBP of N_1 and N_2, and *EBPset* is the set of EBPs iff:

(1) For each $EBP\in EBPset$, then $EBP\leq N_1$ and $EBP\leq N_2$; and

(2) For N', $\exists EBP\in EBPset$ such that $N'\leq EBP$.

Note that an EBP is a subclass of N_1 and N_2. Furthermore, for any a subclass N' of N_1 and N_2, there is an EBP which is a superclass of N', *i.e.* each EBP is a minimal subclass of N_1 and N_2.

Some concepts are introduced to define EBP of labeled WF-nets in the following section.

Definition 14 ***(Pre-transition set and Post-transition set)*** Let N= (P, T, F, L) be a labeled WF-net, then

${}^{\circ}t=\{t' \mid t'\in T\wedge\exists p\in P: (t', p)\in F\wedge(p, t)\in F\}$ is called the pre-transition set of t;

$t^{\circ}=\{t' \mid t'\in T\wedge\exists p\in P: (t, p)\in F\wedge(p, t')\in F\}$ is called the post-transition set of t.

If $t_i\in{}^{\circ}t_{i+1}$ ($1\leq i\leq n$-1), then it is denoted as $t_i\in{}^{\circ+}t_n$ (i=1, 2,…, n-1).

If $t_i\in t_{i-1}{}^{\circ}$ ($2\leq i\leq n$), then it is denoted as $t_i\in t_1{}^{\circ+}$ ($2\leq i\leq n$).

Definition 15 ***(projection)*** Let N= (P, T, F, L) be a labeled WF-net. Then N'= (P', T', F', L') is the projection of N from t_i to t_j, denoted $\Omega(N, t_i, t_j)$, where:

$P'\subseteq P$; $T'= \{t \mid t\in t_i{}^{\circ+}\wedge t\in{}^{\circ+}t_j)\}$;

$F'= F\cap((T'\times P')\cup(P'\times T'))$; and

$L'= \{(t, L(t)) \mid t\in T'\}$.

Given a labeled WF-net N= (P, T, F, L), one operation π is defined to get the pre-set or post-set of transitions.

$\forall t\in T$, $\pi_1(t)=\{p \mid p\in P\wedge(p, t)\in F\}$; $\pi_2(t)=\{p \mid p\in P\wedge(t, p)\in F\}$.

To gain an EBP, the mapping of nodes should be found between the business process and the core process, and labels of transitions can be used to establish the mapping of nodes. Given a labeled WF-net of a business process, the business process extension is the inverse of blocking and hiding operation according to the core processes. The construction algorithm can be described as follows.

Algorithm 2 Construct EBPset

Input: a labeled WF-net N_1= (P_1, T_1, F_1, L_1) of a core process, a labeled WF-net N_2= (P_2, T_2, F_2, L_2) of a business process

Output: Θ

```
Begin
S1: If $\exists t_1\in T_1$ and $t_2\in T_2$ such that $L_1(t_1)$ equal to $L_2(t_2)$, then match $t_1$ with $t_2$, meanwhile map $i_1$, $i_2$ to $i$ and map $o_1$, $o_2$ to $o$.
S2: If for $\forall t\in T_1$, $\exists t'\in T_2$, such that $L_1(t)$ equal to $L_2(t')$, then $\Theta=\{N_2\}$, go to S4.
S3: For each $t\in T_1$ such that $L_1(t)\notin L_2$ do
    3.1 For each $t'\in{}^{\circ}t$ {While ($L_1(t)\notin A_2$) {$t\Leftarrow t'$}, $t_{11}\Leftarrow t$}.
    3.2 For each $t'\in{}^{\circ}t$ {While ($L_1(t)\notin A_2$) {$t\Leftarrow t^{\circ}$}, $t_{12}\Leftarrow t$ }.
    3.3 Let $N'$= ($P'$, $T'$, $F'$, $L'$) be $\Omega(N_1, t_{11}, t_{12})$. Let $t_{21}, t_{22}\in T_2$ such that $L_2(t_{21})$ equal to $L_1(t_{11})$ and $L_2(t_{22})$ equal to $L_1(t_{12})$
    3.4 If $t_{21}\in{}^{\circ}t_{22}$ then
            For each $t_{21}$, $N$= ($P$, $T$, $F$, $L$) where
            {$P=P_2\cup P'-\{\pi_2(t_{21})\}$, $T=T_2\cup T'$, $L=L_2\cup L'$, $F=F_2\cup F'\cup\{t_{21}\times\pi_2(t_{11}),\pi_1(t_{12})\times t_{22})\}-\{t_{21}\times\pi_2(t_{21}),\pi_2(t_{21})\times t_{22}\}$}
            $\Theta=\Theta\cup\{N\}$
          Endfor
        Else $N=(P, T, F, L)$ where
            $\{P=P_2\cup P'-\{\pi_2(t_{21})\}, T=T_2\cup T', L=L_2\cup L', F=F_2\cup F'\cup\{t_{21}\times\pi_2(t_{11}),\pi_1(t_{12}{}^{\circ})\times t_{21}{}^{\circ})\}-\{t_{21}\times\pi_2(t_{21}),\pi_2(t_{21})\times t_{21}{}^{\circ}\}\}$. or
            $\{P=P_2\cup P'-\{\pi_1(t_{22})\}, T=T_2\cup T', L=L_2\cup L', F=F_2\cup F'\cup\{{}^{\circ}t_{22}\times\pi_2(t_{11}),\pi_1(t_{12}{}^{\circ})\times t_{22})\}-\{{}^{\circ}t_{22}\times\pi_1(t_{22}),\pi_1(t_{22})\times t_{22}\}\}$. or
            $\{P=P_2\cup P'-\{\pi_2(t_{11}),\pi_1(t_{12})\}, T=T_2\cup T', L=L_2\cup L', F=F_2\cup F'\cup\{\pi_2(t_{21})\times t_{11}{}^{\circ},{}^{\circ}t_{12}\times\pi_1(t_{22})\}-\{\pi_2(t_{11})\times t_{11}{}^{\circ}, {}^{\circ}t_{12}\times\pi_1(t_{12})\}\}$.
          $\Theta=\Theta\cup\{N\}$
        EndIF
      EndFor
    **S4:** Output $\Theta$.
    End
```

Figure 4. The EBPset Construction Algorithm

For a CP, there may be multiple EBPs on the base of a business process by Algorithm 2. In practice, which EBP will be chosen depends on the specific environment requirement. It is proved that Θ is the EBPset of N_1 and N_2 in the following conclusion.

Theorem 3 Let $N_1=(P_1, T_1, F_1, L_1)$ be a labeled WF-net representing a core process and $N_2=(P_2, T_2, F_2, L_2)$ be a labeled WF-net representing a business process. Let $N'=(P', T', F', L')$ be a labeled WF-net such that $N'\leq N_1$ and $N'\leq N_2$. If Θ is obtained by Algorithm 2, then $\forall N\in\Theta$, $N=(P, T, F, L)$ is an EBP of N_1 and N_2.

Proof. (1) From Algorithm 2, N has a source place i such that $^{\bullet}i=\emptyset$ and a sink place o such that $o^{\bullet}=\emptyset$. Each node of N is on a directed path from i to o, *i.e.* N is strongly connected. So, N is a labeled WF-net.

(2)For $\forall N\in\Theta$, let $A_0=A_1-A_2$, $I=\{L(t)|t\in({}^{\circ+}t)^{\circ+}\wedge L(t)\in A_0\}$ and $H=A_0-I$, then $(\tau_I\circ\partial_H(N),[i]\sim(N_1,[i])$. *i.e.* $N\leq N_1$.

$N\leq N_2$ can be proved similarly.

(3)Let $N'\leq N_1$ and $N'\leq N_2$, denoted $\exists I_1$, H_1, I_2 and $H_2\in L'$, such that $(\tau_{I_1}\circ\partial_{H_1}(N'),[i])\sim(N_1,[i])$ and $(\tau_{I_2}\circ\partial_{H_2}(N'),[i])\sim(N_2,[i])$ then $\exists N=(P, T, F, L)\in\Theta$, such that $(\tau_{I_1\cap I_2}\circ\partial_{H_1\cap H_2}(N'), [i])\sim(N,[i])$, *i.e.* $N'\leq N$.

So, N is an EBP of N_1 and N_2 by Definition 13. Furthermore, Θ is the EBPset of N_1 and N_2.

Theorem 4 shows that an EBP obtained by Algorithm 2 preserves soundness.

Theorem 4 Let $N_1=(P_1, T_1, F_1, L_1)$ be a labeled WF-net representing a core process and $N_2=(P_2, T_2, F_2, L_2)$ be a labeled WF-net representing a business process. Let $N=(P, T, F, L)$ be an EBP from N_1 and N_2 by Algorithm 2. If N_1 and N_2 are sound, then N is sound.

Proof. Let $L'=A_1-A_2$, then $\exists I$ and $H\subseteq L'$, such that $(\tau_I\circ\partial_H(N),[i])\sim(N_2,[i])$ (Theorem 3). For t $(L(t)\in I)$, $(t, L(t))$ is added to N_2 in sequence pattern. For t' $(L(t')\in H)$, $(t', L(t'))$ is added to N_2 in xor-split/xor-join pattern. Then an EBP N is obtained by Algorithm 2. All these operations don't influence the soundness of labeled WF-nets. If N is not sound, then N_1 and/or N_2 are not sound. It contradicts with the hypothesis, so N is sound.

A CP and an EBP can preserve the soundness of original labeled WF-nets. So, proposed algorithms are effective and feasible.

4. Application Instance

Many business processes exist to provide business solutions in typical functional areas. And these processes are typically individualized for countries and enterprises [3]. When these enterprises enter other markets, original business processes are often not suitable for the new market rule or local laws. Some work should be carried to acquire suitable business processes for new environments. Tax declaring is taken an example to show the business process adaption due to the extensively applied for enterprises. Figure 5 and Figure 6 respectively describe the declaring tax processes of enterprises in region R_1 and R_2 (all these processes have been simplified). In Figure 5, the process of enterprise E_1 (N_4) is made up of *tax_appl* (tax application), *asse_tax* (self assessment of taxation), *VAT_start* (VAT start), *pay_VAT* (pay valued-added tax), *VAT_comp* (VAT completion), and *tax_comp* (tax completion), and the process of enterprise E_2 (N_5) includes *input_tax* (input tax) and *output_tax* (output tax) instead of *pay_VAT* in N_4. In Figure 6, the process of enterprise E_3 (N_6) includes *pay_res_tax* (pay resource tax) and *pay_VAT_add_tax* (pay VAT additional tax) except the activities in N_1 and N_2. The process of enterprise E_4 (N_7) includes *pay_pro_tax* (pay property tax) other than *pay_res_tax* in N_6.

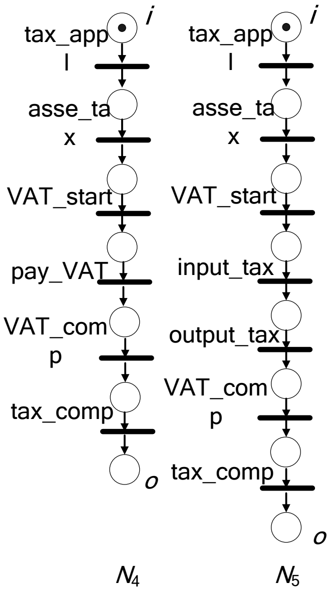


Figure 5. Two Declare Tax Processes in R_1

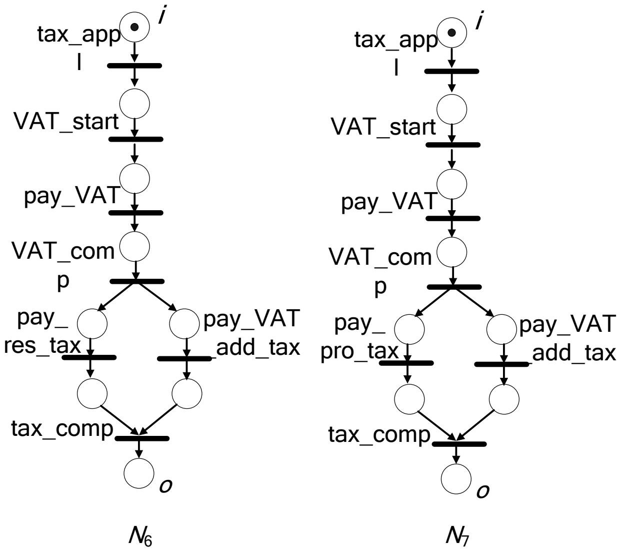


Figure 6. Two Declare Tax Processes in R2

According to the method in [8], the models that share common fragment can be used to construct union of models (called *merged models*). Models N_M and $N_{M'}$ in Figure 7 describe *merged models* of declare tax processes in R_1 and R_2 respectively. *Merged models* can represent and replace original models of the same underlying process. But, when enterprises E_2 of R_1 enter the market of R_2, existing declare tax processes, such as N_2, and *merge model* N_M is not suitable for R_2 because there are extra taxes in declare tax process for R_2. While N_6, N_7 and $N_{M'}$ is appropriate for R_2, they can not describe this situation well due to the activity *asse_tax* of N_5 is ignored. Similarly, configurable processes in [3] can not represent the processes adaption across regions although its functions and connectors are configurable.

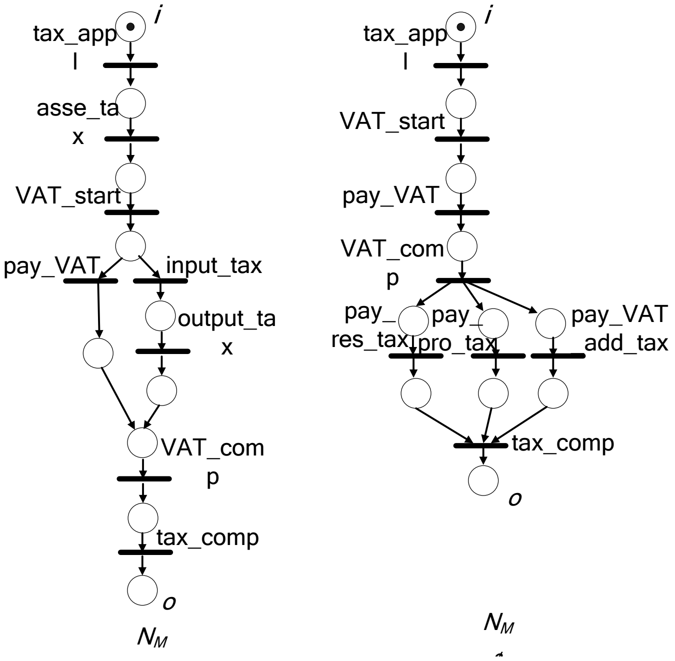


Figure 7. Merged Processes of Declare Tax Processes in R1 and R2

To adapt the process N_5 to meet the requirement of R_2, the core process N_{CP} of N_6 and N_7 can be got according to Algorithm 1. In Figure 8, N_{CP} represents the commonalities of declare tax process in region R_2. Theorems 1 and 2 can be combined to show that N_{CP} is a CP of processes N_6 and N_7, and N_{CP} is sound.

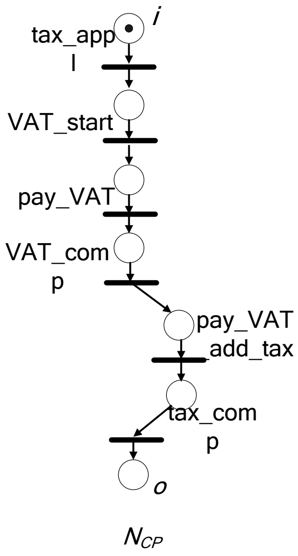


Figure 8. The CP of N_6 and N_7

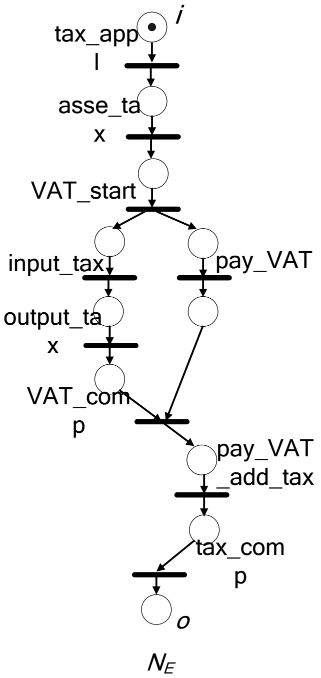


Figure 9. The EBP NE of N5 and NCP

In Figure 9, EBP N_E is acquired based on declare tax processes N_5 and N_{CP} according to Algorithm 2. The activity *pay_*VAT of N_{CP} is similar to activities *input_tax* and *output_tax* of N_2, and they are contained to N_E in a xor-split/xor-join pattern. Meanwhile, N_E contains the *pay_*VAT*_add_tax* of N_{CP} and *asse_tax* of N_5. Thus, N_E reserves the characteristic of N_5 and embodies the commonalities of process in region R_2. Compared with original process N_5 and *merged models* N_M and $N_{M'}$, N_E is more suitable to the specific environment of region R_2. By Theorem 3 and Theorem 4, N_E is sound due to the soundness of N_5 and N_{CP}.

5. Related Works

Business process management is progressively paid attention to in current research work. And it often occurs that multiple variants of a business process exist, and these variants usually originated from different companies or units. How to manage and use these variants is an important part of business process management. Merged model [8] is proposed to converge process variants into a single process. Meanwhile, the unions of process variants (called *merged model*) subsume a collection of process models and intersections of process variants (called *digests*) identify the recurring fragments. Although the commonalities and differences between multiple variants are found, all these ones are applied to replace variants with consolidated model, and process adaption to other environments is not involved.

In some papers, relevant business processes is constructed as a repository (called reference models) which accelerate the modeling process [3, 9]. For example, configurable EPC [3] provides related constraint information of functions and connectors on reference models, and these information make model user can individualise the model by selecting alternative options on a type level. Within each organization, an individual process can be got according to constraint information. Despite diversity of one process among organizations, a process variant can not be applied directly to other regions because the reference model itself is typically individual for countries or enterprises. The commonalities of variant are involved in [6-7]. Common Divisor (MCD) of object life cycle [6] describes the common behavior of objects by hiding and blocking operations and it provides the basis of this paper. But the characteristic and property of MCD are not involved. In [7], core process, the common segment of process variants, is defined from the spectrum of structural and behavioral aspects. Neither algorithm nor further application of the core process is discussed. In this paper, core processes are introduced to adapt the existing process to other regions. And core processes embody the maximal commonalities of the processes in new region. By extending the existing process in the base of core processes, new processes are suitable for the new region reserving the characteristic of existing process. And related theorems prove that core processes and extensional processes are sound.

6. Conclusions

A business process is modelled as a labeled WF-net based on labeled P/T-nets, and the inheritance relation between labeled WF-nets is proposed based on branching bisimilarity and behavioral equivalence. A core process is proposed to describe the commonalities of process variants, and related algorithms are designed and theorems are proved. Existing process model can be extended based on core processes, then extensional business processes are acquired. Extensional business processes has the characteristic of original process model and core processes, so it is suitable to the environment of process variants. The existing methods put emphasis on management of process variants in one region, and extensional business processes are exclusively designed for cross regions in this paper. When an enterprise expands its market to a new region, original processes need adapt themselves to the new economical conditions. We get the new economical rules by researching process variants in the new region and then extend original processes to adapt the new rules. And Extensional business processes can fit the new economical condition well.

Acknowledgements

This work is supported by the National Basic Research Program of China under grant 2010CB328101; the National Natural Science Foundation of China under grants 61170078 and 61173042; the Doctoral Program of Higher Education of the Specialized Research Fund of China under grant 20113718110004; Basic Research Program of Qingdao City of China under grant 13-1-4-116-jch; and the SDUST Research Fund of China under Grant 2011KYTD102.

References

8.1. Journal Article

[2] C. Li, M. Reichert, and A. Wombacher. “The MinAdept Clustering Approach for Discovering Reference Process Models Out of Process Variants”. International Journal of Cooperative Information systems, no. 19 **(2010)**, pp. 3-4.

[3] M. Rosemann, W.M.P. van der Aalst. A configurable reference modelling language. Information Systems. vol. 1, no. 32 **(2007)**.

[4] W.M.P. van der Aalst. “Business process management: A comprehensive survey”. ISRN Software Engineering, 1, 20131 **(2013)**

[5] W.M.P. van der Aalst, M. Dumas, and F. Gottschalk, A. H. M. ter Hofstede, M.La.Rosa, J. Mendling. “Preserving correctness during business process model configuration”, Formal Aspects of Computing, no. 22 **(2010)**, pp. 3-4.

[7] M. Weidlich, M. Weske. “Structural and behavioural commonalities of process variants”. in ZEUS, **(2010)**.

[8] M. La Rosa, M. Dumas, and R. Uba, R. Dijkman. “Business process model merging: an approach to business process consolidation”. ACM Transactions on Software Engineering and Methodology (TOSEM), vol. 2, no. 22 **(2013)**.

[9] F. Gottschalk, W.M.P. van der Aalst, M.H. Jansen-Vullers, H. Monque, and M. La Rosa. “Configurable workflow models”. International Journal of Cooperative Information Systems, vol. 2, no. 17 **(2008)**.

[10] H.M.W. Verbeek, T. Basten, W.M.P. van der Aalst. “Diagnosing workflow processes using Woflan”, The Computer Journal, vol. 4, no. 44 **(2001)**.

[11] W.M.P. Van der Aalst. “The application of Petri nets to workflow management. Journal of circuits”, systems, and computers. vol. 01, no. 8 **(1998)**.

[12] Y.Y. Du, C.J. Jiang, M.C. Zhou, “A Petri net-based model for verification of obligations and accountability in cooperative systems”, IEEE T Syst Man Cy A, vol. 39, no. 2, **(2009)**.

[13] Q. Hu, Y.Y. Du, S.X. Yu, “Service net algebra based on logic Petri nets”, J Inf Sci, no. 268,**(2014)**.

[14] YY Du, LQi, MC Zhou. “A vector matching method for analysing logic Petri nets”. Enterprise Information System.vol. 4, no. 5 **(2011)**.

[16] R.J. Van Glabbeek, W.P. Weijland, “Branching time and abstraction in bisimulation semantics”, Journal of the ACM (JACM), vol. 3, no. 43, **(1996)**.

8.2. Book

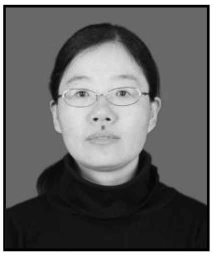
[1] M. Weske, Editor, “Business Process Management: Concepts, Languages, Architectures, Springer-Verlag” , Berlin **(2007)**.

8.3. Conference Proceedings

[6] W.M.P. van der Aalst, T. Basten, “Identifying commonalities and differences in object life cycles using behavioral inheritance”, in: J.M. Colom and M. Koutny(Eds.), Application and Theory of Petri Nets 2001, volume 2075 of Lecture Notes in Computer Science, **(2001)** Springer-Verlag, Berlin.

[15] J. Desel, W. Reisig. Place/transition Petri nets, in W. Reisig and G. Rozenberg(Eds), Lectures on Petri Nets I: Basic Models, volume 1491 of Lecture Notes in Computer Science, **(1998)** Springer-Verlag, Berlin.

Authors



Hongxia Li is a Ph.D. student in School of Information Science and Engineering at Shandong University of Science and technology, China. She received her Master degree in computer science and technology from Shandong University in 2004. Her research focuses on Petri net, process mining and workflow theory.

YuYue Du received the B.S. degree from Shandong University, Jinan, China, in 1982, the M.S. degree from Nanjing University of Aeronautics and Astronautics, Nanjing, China, in 1991, and the Ph.D. degree in computer application from Tongji University, Shanghai, China, in 2003. He is currently a Professor at the College of Information Science and Engineering, Shandong University of Science and Technology, Qingdao, China. He is a member of Professional Committee of Petri Nets of China Computer Federation. His research interests are in formal engineering, Petri nets, real-time systems, Web services, and workflows. He has taken in over ten projects supported by National Nature Science Foundation, National Key Basic Research Developing Program and other important and key projects at provincial levels. He has published more than 140 papers in domestic and international academic publications, and they are embodied more than 80 times by SCI and EI, cited more than 270 times by others.