frontiers | Frontiers in Computer Science

TYPE Original Research
PUBLISHED 13 March 2023
DOI 10.3389/fcomp.2023.1136057

Check for updates

OPEN ACCESS

EDITED BY
Gaiyun Liu,
Xidian University, China

REVIEWED BY
Xiaoliang Chen,
Xihua University, China
Li Yin,
Macau University of Science and Technology,
Macao SAR, China

*CORRESPONDENCE
Dongming Xiang
✉ flysky_xdm@163.com

SPECIALTY SECTION
This article was submitted to Theoretical Computer Science, a section of the journal Frontiers in Computer Science

RECEIVED 02 January 2023
ACCEPTED 20 February 2023
PUBLISHED 13 March 2023

CITATION
Zhao F, Xiang D and Liu G (2023) The trace consistency measurement between WFD-net systems. *Front. Comput. Sci.* 5:1136057. doi: 10.3389/fcomp.2023.1136057

The trace consistency measurement between WFD-net systems

Fang Zhao[1], Dongming Xiang[2]* and Guanjun Liu[3,4]

[1]School of Microelectronics and Data Science, Anhui University of Technology, Maanshan, China, [2]School of Computer Science and Technology, Zhejiang Sci-Tech University, Hangzhou, China, [3]Key Laboratory of the Ministry of Education for Embedded System and Service Computing, Department of Computer Science and Technology, Tongji University, Shanghai, China, [4]Shanghai Electronic Transactions and Information Service Collaborative Innovation Center, Tongji University, Shanghai, China

Introduction: How to measure the trace consistency between process models in workflow net systems have become a crucial part of process mapping, process integration, and difference testing. Workflow net with data (WFD-net) is an effective language to describe a complete model of a workflow net system. Checking their trace consistency is a significant challenge in service-based software engineering. However, most of the existing research on trace consistency measurement has not taken the data into consideration. Therefore, they cannot accurately calculate the trace consistency degree of two WFD-net systems, especially when data operations and guard functions are complex.

Methods: We explore all traces in two WFD-net systems based on guard analysis and data dependencies.

Results: We point out that the trace consistency degree is uncertain when the values of guard functions are considered.

Discussion: We conduct experiments to evaluate our method and show its effectiveness.

KEYWORDS

trace consistency, WFD-net system, data operation, guard analysis, data dependency

1. Introduction

Nowadays, business process models with data are very important in the development of business process systems (Song et al., 2016). They are very effective in reflecting the relations of activities. Here, we mainly focus on studying workflow net systems, which can be used to automate, optimize, and manage these business process models. They have become a major way to construct business processes in software systems (IEEE, 1990; Song et al., 2018). Software designers and business analysts may simulate workflows in the same granularity standards but with different extra activities (i.e., activities without correspondences). Meanwhile, service-based workflow net systems become more and more extensive, and the traces (i.e., firing sequences of transitions) of a system may be affected by data. Therefore, we should provide an effective way to measure the trace consistency of these workflow net systems.

Many models can be taken to describe workflow net systems, i.e., UML activity diagram (Fahland, 2008), Business Process Modeling Notation (BPMN) (van der Aalst et al., 2010; Zhu et al., 2014; Assy et al., 2015), Business Process Execution Language (BPEL) (Weidlich et al., 2014), Event-driven Process Chain (EPC) (Dijkman et al., 2011; Li et al., 2014), and Workflow net with data (WFD-net) (Trcka et al., 2009; Sidorova et al., 2011). Unfortunately, the first four modeling languages have not taken the delete operations into consideration. Compared with these modeling methods, WFD-net is a workflow net with data operations and guard functions. Therefore, we use WFD-net as an adequate formalism to model workflow net systems (Trcka et al., 2009; Sidorova et al., 2011) to analyze the trace consistency in this paper.

At present, many methods are used to measure the trace consistency degree, e.g., DTC (Degree of Trace Consistency) (Weidlich et al., 2011) and CDAT (Consistency Degree of Aligned Traces) (Wang et al., 2018). Based on the sets of projected firing sequences, the first method can be used to obtain the trace consistency degree of two models without data. Based on the direct dependence relations and independence relations on transitions, the second method is used to calculate the trace consistency degree of two traces. Moreover, these methods only considered transitions with correspondence relations, while those without correspondence relations are ignored. However, they may affect the traces of transitions with correspondences. Therefore, these methods are ambiguous and cannot reflect the real traces in two models, especially when there are guard functions and data dependencies. They cannot accurately assess the trace consistency degree of systems with data information. Furthermore, calculating the trace consistency degree would require a lot of work, e.g., if there are k concurrently-executed transitions in a process, then there are $k!$ different traces at most. Therefore, calculating the trace consistency degree would require a lot of work.

To solve these problems, a new behavioral profile relation is defined based on weak behavioral relations, guard functions, control and data dependency. After then, the formalization of correspondence relation and projected firing sequence are provided. Finally, we develop a new method to calculate *trace consistency degree*.

The contributions of this paper are shown in the following:

1) We reveal that the trace consistency degree of two WFD-net systems is non-uniqueness when their guard functions are uncertain since these guard functions may affect the real traces.
2) We point out that transitions without correspondence relations but with data dependencies should not be ignored as they may affect the traces of the systems.
3) We minimize the number of traces in WFD-net systems based on guard functions and data dependencies.

The rest of this paper is arranged as follows. Section 2 introduces some basic concepts used in this paper and provides a motivating example. Section 3 defines control dependency, data dependency and behavioral profile relations, order relation, firing sequence, trace, correspondence relation, projected firing sequence and trace consistency measurement method. Section 4 presents experiments to show that guard functions and data dependencies can affect the traces in WFD-net systems. Section 5 reviews the related work about the existing consistency measurement methods. Section 6 concludes this paper and presents future work.

2. Preliminaries and motivation

In this section, we first introduce some concepts about WF-net, weak order relation, weak behavioral profile, and WFD-net. Some symbols are demonstrated in Table 1. The related details can be found in Li and Liu (2009), Weidlich and van der Werf (2012), Liu et al. (2017), Liu et al. (2018), Xiang et al. (2021), and Zhao et al. (2022a). Furthermore, a motivating example is given.

TABLE 1 Common symbols used in this paper.

Symbols	Descriptions
$\mathbb{N}$	Set of positive integers
BP	Weak behavioral profile
MD	Missing data
BP	Behavioral profile
DTC	Degree of trace consistency (Weidlich et al., 2011)
TCDW	Trace consistency degree of WFD-net systems

2.1. Preliminaries

A Petri net $N = (P, T, F)$ as a tool is often used to simulate and analyze the trace (i.e., firing sequences of transitions) of a workflow system (Liu et al., 2019; Zhang et al., 2022), where P is a finite non-empty set of places, T is a finite non-empty set of transitions, $P \cap T = \varnothing$, and $F \subseteq (P \times T) \cup (T \times P)$ is a set of arcs. Given a node $x \in P \cup T$, the preset of x is ${}^{\cdot}x = \left\{y \middle| \left(y \in P \cup T\right) \cap \left(y, x\right) \in F\right\}$, and the post-set of x is $x^{\cdot} = \{y|(y \in P \cup T) \cap (x, y) \in F\}$. If x is the initial place, it must satisfy ${}^{\cdot}x = \varnothing$; If x is the final place, it must satisfy $x^{\cdot} = \varnothing$. Given a set of nodes $X \subseteq P \cup T$ and $x \in X$, the preset of X is denoted by ${}^{\cdot}X$, and the post-set of X is denoted by $X^{\cdot}$, where ${}^{\cdot}X = \{Y|(Y \subseteq P \cup T) \wedge (y \in Y) \wedge (y, x) \in F\}$ and $X^{\cdot} = \{Y|(Y \subseteq P \cup T) \wedge (y \in Y) \wedge (x, y) \in F\}$. In Figure 1A, if $X = \{t_5, t_6\}$, we have ${}^{\cdot}X = {}^{\cdot}t_5 \cup {}^{\cdot}t_6 = \{p_4, p_5\}$, and $X^{\cdot} = t_5^{\cdot} \cup t_6^{\cdot} = \{p_5, p_7\}$. If we use k to denote the preset number, then we have ${}^{(\cdot)_1}t_5 = {}^{\cdot}t_5 = \{p_4\}$, and $k = 1$ (resp. ${}^{(\cdot)_3}t_6 = {}^{\cdot}({}^{\cdot}({}^{\cdot}t_6)) = \{p_4\}$, and $k = 3$). For the post-set number of X, it can be defined analogously.

The following parts present the workflow net (WF-net), as a special Petri net satisfying the following conditions (Liu et al., 2013).

Definition 1 [WF-net (Liu, 2014, 2022)]. Let $N = (P, T, F)$ be a workflow net (WF-net), if the following conditions hold:

(1) There exists only one place $p_i \in P$ as begin and only one place $p_o \in P$ as end in N; and
(2) The short-circuit net $N^* = (P, T \cup \left\{t'\right\}, F \cup \left\{\left(p_o, t'\right), \left(t', p_i\right)\right\})$ is strongly connected, where $t' \notin T$.

A WF-net system (N, M_0) is a WF-net extended with an initial marking M_0. The dynamic behaviors of workflow systems are decided by distributing different markings. Figures 1A, B depict two WF-net systems if the guard functions and data operations are ignored.

In general, we can distinguish three basic weak behavioral relations between transition pairs in a WF-net system, and they are based on the concept of *weak order relation* (Weidlich et al., 2011).

Definition 2 [Weak order relation (Weidlich et al., 2011; Zhao et al., 2022b)]. Given a WF-net system $(N, M_0) = (P, T, F, M_0)$, the weak order relation $\succ_c \subset T \times T$ contains transition pairs $t_x, t_y \in T$, there exists a weak firing sequence $\sigma = t_1 t_2 \cdots t_n (n \in \mathbb{N})$ with $(N, M_0)\,[\sigma >, x \in \{1, 2, \cdots, n-1\}$, and $x < y \leq n$.

For the example of Figure 1A, we have $t_1 \succ_c t_2$ and $t_2 \not\succ_c t_1$. The set of all weak firing subsequences is denoted by T^*, and

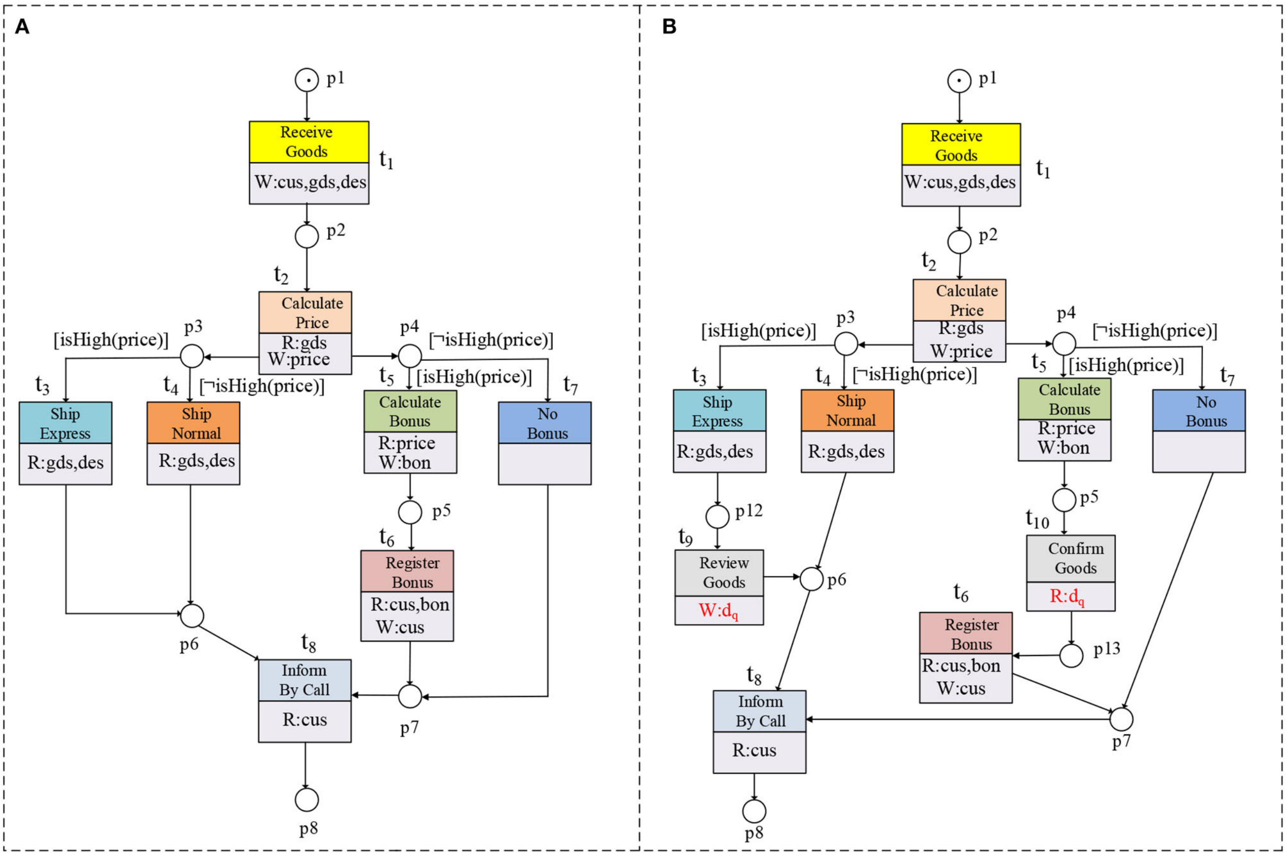

FIGURE 1
Two shipper process models are shown in **(A, B)**. There are no correspondences of t_9, t_{10}, and d_q in **(B)**.

$T^* = \{\sigma_{(1)}\} \cup \{\sigma_{(2)}\} \cup \cdots \cup \{\sigma_{(\varepsilon)}\}$, where $\sigma_{(\varepsilon)} = t_{x_1} t_{x_2} \cdots t_{x_\varepsilon}$, and $t_{x_1}, t_{x_2}, \cdots, t_{x_\varepsilon} \in T$.

Based on the above definitions, *weak behavioral profile* between two transitions is shown in the following.

Definition 3 [Weak behavioral profile, bp (Weidlich et al., 2011)]. Given a WF-net system $(N, M_0) = (P, T, F, M_0)$, a pair of transitions $t_x, t_y \in T$ is in one of the following relations, where $k \in \{1, 3, \cdots, n\}, n = 2m - 1$, and $m \in \mathbb{N}$:

(1) Weak strict order relation $(t_x \rightarrow_c t_y)$, if $(t_x \succ_c t_y) \wedge (t_y \not\succ_c t_x)$;
(2) Weak exclusiveness relation $(t_x +_c t_y)$, if $(t_x \not\succ_c t_y) \wedge (t_y \not\succ_c t_x)$; or
(3) Weak interleaving order relation $(t_x \parallel_c t_y)$, if $(t_x \succ_c t_y) \wedge (t_y \succ_c t_x)$. It contains the following two kinds:
   1) Weak circulation relation $(t_x \leftrightarrow_c t_y)$, if $(t_x \succ_c t_y) \wedge (t_y \succ_c t_x) \wedge (t_x^{\cdot} \cap (\cup_{k=1}^{n} {}^{(\cdot)k} t_y) \neq \varnothing) \wedge (t_y^{\cdot} \cap (\cup_{k=1}^{n} {}^{(\cdot)k} t_x) \neq \varnothing)$; or
   2) Weak concurrency relation $(t_x \rightleftarrows_c t_y)$, if $(t_x \succ_c t_y) \wedge (t_y \succ_c t_x) \wedge (t_x^{\cdot} \cap (\cup_{k=1}^{n} {}^{(\cdot)k} t_y) = \varnothing) \wedge (t_y^{\cdot} \cap (\cup_{k=1}^{n} {}^{(\cdot)k} t_x) = \varnothing)$.

The above relations comprise the weak behavioral profile of (N, M_0), denoted as $bp = \{bp_1, bp_2, bp_3\}$, where $bp_1 = \rightarrow_c$, $bp_2 = +_c$, and $bp_3 = \parallel_c$. As for the inverse order relation of bp_1 [i.e., $(t_x \not\succ_c t_y) \wedge (t_y \succ_c t_x)$], we denote it as $\rightarrow_c^{(-1)}$ or bp_{-1}. If $t_x \rightarrow_c t_y$, then we have $t_y \rightarrow_c^{(-1)} t_x$ or $(t_y, t_x) = bp_{-1}$. The weak behavioral profile relation is also called weak behavioral relation in this paper. In Figure 1A, we have $t_1 \rightarrow_c t_2$, $t_3 +_c t_4$, and $t_3 \parallel_c t_5$. They are denoted as $(t_1, t_2) = bp_1$, $(t_3, t_4) = bp_2$, and $(t_3, t_5) = bp_3$, respectively.

A workflow net with data (*WFD-net*) is a WF-net with every transition $t \in T$ labeled with four elements at most:

(1) Some data items need to be read before firing t;
(2) Some data items would be written after firing t;
(3) Some data items may be deleted after firing t;
(4) Some data items may be the guard functions of t (optional).

The update of data items can be seen as the combination of read/write operations on transitions. Furthermore, writing a data item on a transition for the first time can be known as creating the data item. We also assume that there are no read operations or guard functions on the initial transition (Trcka et al., 2009; Sidorova et al., 2011).

Definition 4 [WFD-net (Trcka et al., 2009; Sidorova et al., 2011)]. A WFD-net $ND = (P, T, F, D, Guard, R, W, De, Gd)$ is a workflow net (WF-net) extended with data, it consists of the following:

(1) (P, T, F) is a WF-net;
(2) D is a different set of data items;
(3) $Guard$ is a set of guards over D;

(4) $R: T \rightarrow 2^D$ is the read operation set;
(5) $W: T \rightarrow 2^D$ is the write operation set;
(6) $De: T \rightarrow 2^D$ is the delete operation set; and
(7) $Gd: T \rightarrow Guard$ is a function that assign guards to transitions.

A WFD-net system is a WFD-net with an initial marking M_0, it can be denoted as $(ND, M_0) = (P, T, F, M_0, D, Guard, R, W, De, Gd)$. In the following, all workflow systems are in the form of WFD-net systems. Figures 1A, B depict two WFD-net systems.

2.2. Motivating example

This subsection takes two *shipper process models* as a motivating example to illustrate our work, as shown in Figures 1A, B. There are 5 same data items in these figures (i.e., *cus*, *gds*, *des*, *price*, and *bon*, where *cus* denotes the data of customers, *gds* stands for the data of goods, *des* represents the destination of goods, *price* means the price of goods, and *bon* denotes the bonus), and the set of guards is $Guard = \{isHigh\left(price\right), \neg isHigh\left(price\right)\}$. For the transition t_5, its data operations and guard function are $R\left(t_5\right) = \{price\}$, $W\left(t_5\right) = \{bon\}$, $De\left(t_5\right) = \varnothing$, and $Gd(t_5) = \{isHigh\left(price\right)\}$, respectively.

Figure 1A describes a shipper process model in the shape of a WFD-net system. Firstly, shippers receive goods from customers and write 3 data items, i.e., the data of customers and goods, and the destination of goods. Then, shippers should calculate the price by reading the data of goods and generating the price. Afterwards, customer advisers and shipment departments execute their tasks concurrently. If the goods' price is high, express shipment is used by reading the data of goods and the destination of the goods. Under the same condition, customer advisors calculate the bonus for customers by reading the price and producing the bonus. If the price of goods is not high, normal shipment is taken by reading the data of goods and the destination of goods, and there is no bonus to calculate. If the customer advisors calculate the bonus for customers, then they should register the bonus by reading the data of customers and the bonus as well as generating new data of customers. At last, customers who send goods are informed by call.

Figure 1B shows a similar model to Figure 1A. However, there are some extra transitions (i.e., t_9 and t_{10}) and data items (i.e., d_q) without any correspondences in Figure 1B. The transition t_9 represents *Review Goods*, the transition t_{10} denotes *Confirm Goods*, and the data item d_q stands for *the quality of the goods*.

The method of DTC ignores those transitions without correspondence, e.g., t_9 and t_{10} in Figure 1B. Therefore, the method of DTC is insensitive to these extended elements, and the trace consistency degree of Figures 1A, B computed by this method is 1. This result means that the traces in the above figures are the same. However, these extensions may produce data dependencies (cf. Def. 6) and affect the real traces. Therefore, the DTC method may be inappropriate for analyzing the real trace consistency degrees of WFD-net systems.

3. Trace consistency measurement based on data and behavioral profiles

Before introducing *the trace consistency degree of two WFD-net systems*, we present *control dependency* and *data dependency* of transition pairs in this section. Then, we illustrate trace and trace consistency degree.

Definition 5 (Control dependency). Given two transitions t_x and t_y in (ND, M_0), $t_y$is control-dependent on t_x if $t_x \rightarrow_c t_y$.

When the specific read/delete operation of a transition is derived from the write operation of another transition, we say there exists a data dependency. The data dependency may affect the traces in a model. The specific details are shown as follows.

Definition 6 (Data dependency). Given two transitions t_x and t_y in (ND, M_0), t_y is data-dependent on t_x (i.e., $t_x \rightarrow_c t_y$) if:

(1) There exists a data item $d \in D: \left[d \in W\left(t_x\right) \cap \left(R\left(t_y\right) \cup Gd\left(t_y\right)\right)\right] \wedge \left[d \notin De\left(t_x\right)\right]$; and
(2) There is no transition t_w enabled between t_x and t_y, such that $d \in W\left(t_w\right) \cup De\left(t_w\right)$.

Let t_x and t_y be a pair of transitions derived from (ND, M_0). The control dependency indicates there is a weak strict order relation such that $t_x \rightarrow_c t_y$. In Figures 1A, B, we have $t_1 \rightarrow_c t_2$. The control dependency satisfies the transitivity. That is, if $(t_x \rightarrow_c t_y) \wedge (t_y \rightarrow_c t_r)$, then we have $t_x \rightarrow_c t_r$. A transition t_y is data-dependent on another transition t_x implies that t_y reads/deletes a data item written by t_x. If there is no data dependency between t_x and t_y, then we have $t_x \parallel_d t_y$. In Figure 1B, t_{10} is data-dependent on t_9 as $d_q \in W(t_9) \cap R(t_{10})$ and there is no transition t_w enabled between t_9 and t_{10}, such that $d_q \in W(t_w) \cap De(t_w)$.

Definition 7 [Distance between transitions in weak circulation relation (Zhao et al., 2022a)]. Let $(ND, M_0) = (P, T, F, M_0, D, Guard, R, W, De, Gd)$ be a WFD-net system, $\exists t_x, t_y \in T: t_x \leftrightarrow_c t_y$, $t_x \in t^{(\cdot)k_1}$, and $t_y \in t^{(\cdot)k_2}$, where t is a transition outside the weak circulation relation with t_x and t_y, k_1 and k_2 are the minimum number of post-sets, such that $k_1, k_2 \in \{2, 4, \cdots, 2m\}$, and $m \in \mathbb{N}$. The distance between t_x and t_y is defined as $\delta\left[t_y, t_x\right] = k_2 - k_1$.

For the example of Figure 2A, we have $t_x \leftrightarrow_c t_y$, and $\delta\left[t_y, t_x\right] = k_2 - k_1 = 2$.

3.1. Transition pairs behavioral relations in a WFD-net system

Before formalizing the relation of transitions t_x and t_y in a WFD-net system, we first explain some relevant symbols as follows:

(1) $Rel(Gd\left(t_x\right))$ (resp. $Rel(Gd\left(t_y\right))$) is the set of data items in guards related to t_x (resp. t_y), and $Rel(Gd\left(t_x(d)\right))$ (resp. $Rel(Gd\left(t_y(d)\right))$) is the value of d in guards related to t_x (resp. t_y); and
(2) $\Gamma(Rel\left(Gd\left(t_x\right)\right))$ (resp. $\Gamma(Rel\left(Gd\left(t_y\right)\right))$) is the set of final values of data items in $Rel\left(Gd\left(t_x\right)\right)$(resp. $Rel(Gd\left(t_y\right))$), and $\Gamma(Rel\left(Gd\left(t_x(d)\right)\right))$(resp. $\Gamma(Rel\left(Gd\left(t_y(d)\right)\right))$ is the final value of d in $Rel\left(Gd\left(t_x\right)\right)$(resp. $Rel(Gd\left(t_y\right))$).

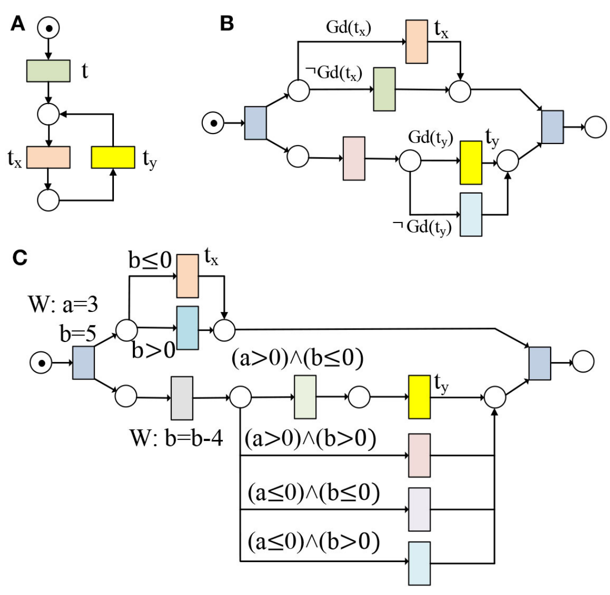

FIGURE 2
(A) t_x and t_y are in weak circulation relation (i.e., $t_x \leftrightarrow_c t_y$); **(B, C)** The behavioral relation of t_x and t_y depends on their guard functions.

For the transition t_y in Figure 2C, we have

(1) $Rel(Gd(t_y))$ is $\{a, b\}$, and $Rel(Gd(t_y(a)))$(resp. $Rel(Gd(t_y(b)))$ is $a > 0$ (resp. $b \le 0$) in guards related to t_y; and
(2) $\Gamma(Rel(Gd(t_y)))$ is the set of final values of $\{a, b\}$ in $Rel(Gd(t_y))$, and $\Gamma(Rel(Gd(t_y(a))))$(resp. $\Gamma(Rel(Gd(t_y(b))))$ is $a = 3$ (resp. $b = 5 - 4 = 1$) in $Rel(Gd(t_y))$.

The transition pairs' behavioral relations in a WFD-net system depend on their weak behavioral relations, data dependencies, and guard functions. The following behavioral relations compose the *behavioral profile* of a WFD-net system.

Definition 8 (Behavioral profile, BP). Given a WFD-net system $(ND, M_0) = (P, T, F, M_0, D, Guard, R, W, De, Gd)$ and a pair of transitions t_x,$t_y \in T$, we have

(1) Strict order relation ($t_x \rightarrow_{cd} t_y$), if
   1) $(t_x \rightarrow_c t_y) \wedge \left[(t_x \rightarrow_d t_y) \vee (t_x \parallel_d t_y)\right]$; or
   2) $(t_x \rightleftarrows_c t_y) \wedge (t_x \rightarrow_d t_y)$.
(2) Exclusiveness relation ($t_x +_{cd} t_y$), if
   1) $(t_x +_c t_y) \wedge (t_x \parallel_d t_y)$;
   2) If $(t_x \parallel_d t_y) \vee (t_x \rightarrow_c t_y) \vee (t_y \rightarrow_c t_x)$, $\exists d : d \in Rel(Gd(t_x)) \cap Rel(Gd(t_y))$, and $\left[Rel(Gd(t_x(d))) \cap \Gamma(Rel(Gd(t_x(d)))) = \varnothing\right] \vee \left[Rel(Gd(t_y(d))) \cap \Gamma(Rel(Gd(t_y(d)))) = \varnothing\right]$;
   3) If $t_x \rightleftarrows_c t_y$, $\exists d : d \in Rel(Gd(t_x)) \cap Rel(Gd(t_y))$, and $Rel(Gd(t_x(d))) \cap Rel(Gd(t_y(d))) = \varnothing$.
   or
(3) Interleaving order relation ($t_x \parallel_{cd} t_y$). It can be divided into the following two kinds:
   1) Circulation relation ($t_x \leftrightarrow_{cd} t_y$), if $(t_x \leftrightarrow_c t_y) \wedge \left[(t_x \parallel_d t_y) \vee ((\delta[t_y, t_x] \ge 0) \wedge (t_x \rightarrow_d t_y))\right]$; or
   2) Concurrency relation ($t_x \rightleftarrows_{cd} t_y$), if
      i) $t_x \rightleftarrows_c t_y$ and $\nexists d : d \in Rel(Gd(t_x)) \cap Rel(Gd(t_y))$; or
      ii) $t_x \rightleftarrows_c t_y$, $Rel(Gd(t_x)) \cap Rel(Gd(t_y)) = \{\mathfrak{D}\}$,and $\forall d \in \mathfrak{D} : [Rel(Gd(t_x(d))) \cap \Gamma(Rel(Gd(t_x(d)))) \neq \varnothing] \wedge [Rel(Gd(t_y(d))) \cap \Gamma(Rel(Gd(t_y(d)))) \neq \varnothing]$∘

These relations constitute the behavioral profile of (ND, M_0), denoted as $BP = \{BP_1, BP_2, BP_3\}$, where $BP_1 = \rightarrow_{cd}$, $BP_2 = +_{cd}$, and $BP_3 = \parallel_{cd}$. We use BP_{-1} or $\rightarrow_{cd}^{-1}$ to show the inverse order relation of BP_1. It's worth noting that each transition pair can be in a single behavioral relation.

As we know, $Rel(Gd(t_3(price))) = Rel(Gd(t_5(price))) = Rel(Gd(t_6(price))) = \{[isHigh(price)]\}$ and $Rel(Gd(t_4(price))) = Rel(Gd(t_7(price))) = \{[\neg isHigh(price)]\}$. Therefore, we have $Rel(Gd(t_3(price))) \cap Rel(Gd(t_7(price))) = \varnothing$, $Rel(Gd(t_4(price))) \cap Rel(Gd(t_5(price))) = \varnothing$, and $Rel(Gd(t_4(price))) \cap Rel(Gd(t_6(price))) = \varnothing$. According to Def. 8, we have $(t_3, t_7) = (t_4, t_5) = (t_4, t_6) = BP_2$, and $(t_3, t_5) = (t_4, t_7) = BP_3$ in Figures 1A, B after taking these guard functions into consideration. Without considering data dependencies of transitions without correspondences, we have $(t_3, t_6) = BP_3$ in Figures 1A, B. After considering the data item d_q, we know $d_q \in W(t_9) \cap R(t_{10})$, and $t_9 \rightarrow_{cd} t_{10}$ as t_{10} is data-dependent on t_9 in Figure 1B. Therefore, we have $(t_3, t_6) = BP_3$ in Figure 1A and $(t_3, t_6) = BP_1$ in Figure 1B. In Figure 2B, if there exists no data item d satisfying $d \in Rel(Gd(t_x)) \cap Rel(Gd(t_y))$, then we have $t_x \rightleftarrows_{cd} t_y$.

Definition 9 (Order relation). Let $(ND, M_0) = (P, T, F, M_0, D, Guard, R, W, De, Gd)$ be a WFD-net system, the order relation $t_x \succ t_y$ includes transitions t_x and t_y satisfying one of the following conditions:

(1) $t_x \rightarrow_c t_y$;
(2) $t_x \rightleftarrows_c t_y$;
(3) $(t_x \leftrightarrow_c t_y) \wedge (\delta[t_y, t_x] \ge 0)$.

If some transitions lack data dependencies, there is a *missing data*. Based on the above relation, we further formalize missing data.

Definition 10 (Missing data, MD). Given a WFD-net system $(ND, M_0) = (P, T, F, M_0, D, Guard, R, W, De, Gd)$, T^* is the set of all weak firing subsequences, $\exists \sigma \in T^*$, and $\exists t_m \in \sigma : p_i \in t_m$. A data item $d \in D$ is missing if one of the following conditions is satisfied:

(1) $\exists t_y \in T$, $\forall t_x \in \sigma : (t_x \succ t_y) \wedge [((d \notin W(t_x)) \wedge (d \in De(t_x) \cup R(t_y))) \vee ((d \notin W(t_x) \cup W(t_y)) \wedge (d \in De(t_y)))]$, which is denoted by $d \lessdot MD$; or
(2) $\exists t_x \in \sigma, \exists t_z \in T, \forall t_y \in T : (t_x \succ t_z) \wedge (t_x \succ t_y) \wedge (t_y \succ t_z) \wedge [d \in W(t_x) \cap (De(t_x) \cup De(t_y)) \cap (R(t_z) \cup De(t_z))] \wedge (d \lessgtr R(t_y)) \wedge [d \notin W(t_y) \cup W(t_z)]$, which is denoted by $d \lessdot MD$, where $d \lessgtr R(t_y)$ denotes $d \in R(t_y)$ or $d \notin R(t_y)$.

A missing data appears when a data item is read or deleted but not written before (Meda et al., 2010), as shown in Figures 3A–D. This error may affect the trace (Def. 12) in a WFD-net system. If the missing data is not an ending error, we can analyze the trace consistency of two WFD-net systems. As shown in Figure 3B, we have $t_3 \rightarrow_{cd} t_2$ as t_2 is data-dependent on t_3.

Notice that, if $[(t_x \rightarrow_c t_y) \wedge (t_y \rightarrow_d t_x)] \vee [(t_x +_c t_y) \wedge ((t_x \rightarrow_d t_y) \vee (t_y \rightarrow_d t_x))] \vee [(t_x \leftrightarrow_c t_y) \wedge (\delta[t_y, t_x] > 0) \wedge (t_y \rightarrow_d t_x)]$, there is a missing data. When the missing data is an ending error, there is no point in analyzing the traces in two WFD-net models.

3.2. Firing sequence and traces in a WFD-net System

Before defining the *trace*, we present the definition of *firing sequence*.

Definition 11 (Firing sequence). Given a WFD-net system $(ND, M_0) = (P, T, F, M_0, D, Guard, R, W, De, Gd)$, $\sigma_i = t_1 t_2 \cdots t_n (n \in \mathbb{N})$ is a firing sequence in (ND, M_0) if

(1) $t_x \rightarrow_{cd} t_y$, where $1 \leq x < y \leq n$; or
(2) $t_x \parallel_{cd} t_y$, where $1 \leq x < y \leq n$.

If $t_x \rightarrow_{cd} t_y$, then we denote it as $t_x t_y$. If $t_x \parallel_{cd} t_y$, then we denote it as $\{(t_x)(t_y)\}$. If $t_x \parallel_{cd} t_y$, $t_x \parallel_{cd} t_r$, and $t_y \rightarrow_{cd} t_r$, then we denote it as $\{(t_x)(t_y t_r)\}$. As shown in Figure 1A, $t_1 t_2\{(t_4)(t_7)\}t_8$ is a firing sequence.

Definition 12 (Trace). Given a WFD-net system $(ND, M_0) = (P, T, F, M_0, D, Guard, R, W, De, Gd)$, a trace in (ND, M_0) is a firing sequence $\sigma_i = t_1 t_2 \cdots t_n (n \in \mathbb{N})$. The set of all traces is defined as $\varphi = \cup_{i=1}^{k}\{\sigma_i\}$, where k is the number of traces.

The definition of a trace depends on the firing sequence, and there is no circulation relation by default. For the example of Figure 1A, the traces are $t_1 t_2\{(t_3)(t_5 t_6)\}t_8$ and $t_1 t_2\{(t_4)(t_7)\}t_8$. After then, we provide a way to compute the trace consistency degree of two WFD-net systems.

3.3. Trace consistency measurement method

Before providing the trace consistency measurement method, we must introduce *correspondence relation* and *projected firing sequence* in WFD-net systems. The correspondence relation of two WFD-net systems relates their transitions to each other. In the following, we formalize the correspondence relation.

Definition 13 [Correspondence relation (Weidlich et al., 2011)]. Let $(ND_1, M_{01}) = (P_1, T_1, F_1, M_{01}, D_1, Guard_1, R_1, W_1, De_1, Gd_1)$ and $(ND_2, M_{02}) = (P_2, T_2, F_2, M_{02}, D_2, Guard_2, R_2, W_2, De_2, Gd_2)$ be two WFD-net systems. The non-empty correspondence relation $\sim \subseteq T_1 \times T_2$ aligns (ND_1, M_{01}) and (ND_2, M_{02}) by relating corresponding transitions to each other. $T_1^{\sim} \subseteq T_1$ is a set of transitions with correspondences in (ND_1, M_{01}) satisfying $T_1^{\sim} = \{t_x \in T_1 | \exists t_y \in T_2 : t_x \sim t_y, R(t_x) = R(t_y), W(t_x) = W(t_y), De(t_x) = De(t_y), Gd(t_x) = Gd(t_y)\}$. The set $T_2^{\sim}$ for (ND_2, M_{02}) is defined analogously.

For the examples of Figures 1A, B, the set of transitions with correspondences in Figure 1A is $T_1^{\sim} = \{t_1, t_2, t_3, t_4, t_5, t_6, t_7, t_8\}$.

Definition 14 [Projected firing sequence (Weidlich et al., 2011)]. Let $(ND_1, M_{01}) = (P_1, T_1, F_1, M_{01}, D_1, Guard_1, R_1, W_1, De_1, Gd_1)$ and $(ND_2, M_{02}) = (P_2, T_2, F_2, M_{02}, D_2, Guard_2, R_2, W_2, De_2, Gd_2)$ be two WFD-net systems. The projected firing sequence $\sigma^{\sim}$ is a firing sequence that contains all transitions with correspondences.

As shown in Figure 1B, $t_1 t_2\{(t_3)(t_5)\}t_6 t_8$ is a projected firing sequence.

Definition 15 (Trace consistency degree of WFD-net systems, TCDW). Let $(ND_1, M_{01}) = (P_1, T_1, F_1, M_{01}, D_1, Guard_1, R_1, W_1, De_1, Gd_1)$ and $(ND_2, M_{02}) = (P_2, T_2, F_2, M_{02}, D_2, Guard_2, R_2, W_2, De_2, Gd_2)$ be two WFD-net systems. They are aligned by $\sim$. Let $\varphi_1^{\sim}$ and $\varphi_2^{\sim}$ be the sets of all projected firing sequences, such that $\sigma_1^{\sim} \in \varphi_1^{\sim}$ and $\sigma_2^{\sim} \in \varphi_2^{\sim}$ be the sets of same projected firing sequences. The trace consistency degree is defined as $TCDW^{\sim} = \frac{|\sigma_1^{\sim}|+|\sigma_2^{\sim}|}{|\varphi_1^{\sim}|+|\varphi_2^{\sim}|}$.

3.4. Trace consistency degree of WFD-net systems

Notice that there are only 1:1 correspondence relation in this paper. The transitions with correspondence relations read, write and delete the same data items. Therefore, we use the same transitions to denote them (e.g., transitions $t_1 \sim t_8$ in Figures 1A, B). A trace consistency guarantees that the essential

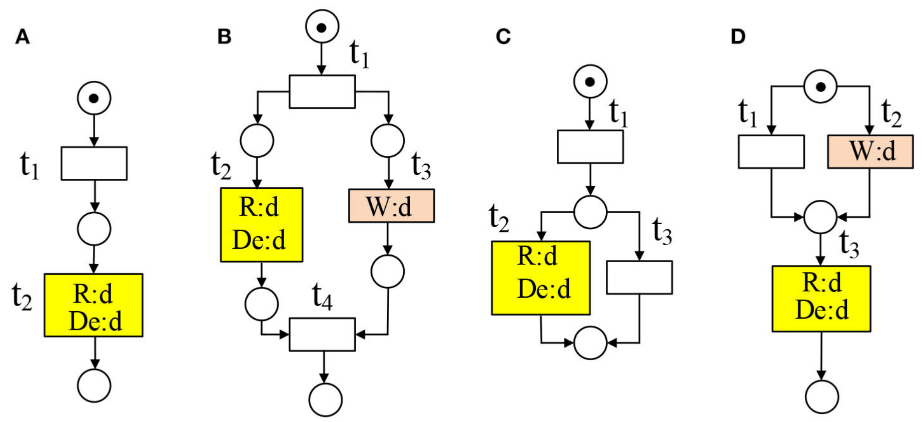

FIGURE 3
d is a missing data in **(A–D)**.

Input: Two WFD-net systems (ND_1, M_{01}) and (ND_2, M_{02}) with the sets of projected firing sequences $\varphi_1^{\sim}$ and $\varphi_2^{\sim}$, and the sets of same projected firing sequences $\sigma_1^{\sim}$ and $\sigma_2^{\sim}$.
Output: Trace consistency degree of two WFD-net systems: $TCDW^{\sim}$

1: $TCDW^{\sim} = \varnothing$, $\sigma_1^{\sim} = \varnothing$, $\sigma_2^{\sim} = \varnothing$, $\varphi_1^{\sim} = \varnothing$, and $\varphi_2^{\sim} = \varnothing$;
2: **for** $\varphi_a \in \varphi_1$ and $\exists t_i \in \varphi_a : t_i \notin \varphi_2$ **do**
3: $\varphi_1^{\sim} = \varphi_1^{\sim} \cup (\frac{\varphi_a}{t_i})$
4: **end for**
5: **for** $\varphi_b \in \varphi_2$ and $\exists t_j \in \varphi_b : t_j \notin \varphi_1$ **do**
6: $\varphi_2^{\sim} = \varphi_2^{\sim} \cup (\frac{\varphi_b}{t_j})$
7: **end for**
8: **for** $\forall \sigma_c^{\sim} \in \varphi_1^{\sim}$, $\exists \sigma_d^{\sim} \in \varphi_2^{\sim}$ **do**
9: **if** $\sigma_c^{\sim} = \sigma_d^{\sim}$ **then**
10: $\sigma_1^{\sim} = \sigma_1^{\sim} \cup \sigma_c^{\sim}$;
11: **else**
12: $\sigma_1^{\sim} = \sigma_1^{\sim}$;
13: **end if**
14: **end for**
15: **for** $\forall \sigma_e^{\sim} \in \varphi_2^{\sim}$, $\exists \sigma_f^{\sim} \in \varphi_1^{\sim}$ **do**
16: **if** $\sigma_e^{\sim} = \sigma_f^{\sim}$ **then**
17: $\sigma_2^{\sim} = \sigma_2^{\sim} \cup \sigma_f^{\sim}$;
18: **else**
19: $\sigma_2^{\sim} = \sigma_2^{\sim}$;
20: **end if**
21: **end for**
22: $TCDW^{\sim} = \frac{|\sigma_1^{\sim}|+|\sigma_2^{\sim}|}{|\varphi_1^{\sim}|+|\varphi_2^{\sim}|}$;

Algorithm 1. Trace consistency degree of two WFD-net systems.

rotes are reserved among transitions with correspondences in two WFD-net systems. According to our method, the inconsistencies manifest there exist traces without correspondence.

According to Def. 15, we propose Algorithm 1 to calculate the trace consistency degree. This algorithm uses the sets of projected firing sequences $\varphi_1^{\sim}$ and $\varphi_2^{\sim}$, and the sets of the same projected firing sequences $\sigma_1^{\sim}$ and $\sigma_2^{\sim}$ in two WFD-net systems as inputs, and the trace consistency degree $TCDW^{\sim}$ as outputs. Step 1 initializes $TCDW^{\sim}$, $\varphi_1^{\sim}$, $\varphi_2^{\sim}$, $\sigma_1^{\sim}$, and $\sigma_2^{\sim}$. $\varphi_1^{\sim}$ and $\varphi_2^{\sim}$ can be calculated by taking Steps 2–7. $\sigma_1^{\sim}$ and $\sigma_2^{\sim}$ are derived from Steps 8–21. Finally, the trace consistency degree $TCDW^{\sim}$ is calculated according to $\varphi_1^{\sim}$, $\varphi_2^{\sim}$, $\sigma_1^{\sim}$, and $\sigma_2^{\sim}$, as shown in Step 22.

The number of firing sequences in (ND_1, M_{01}) is n_1, and the number of firing sequences in (ND_2, M_{02}) is n_2. The time cost of finding projected firing sequences in (ND_1, M_{01}) and (ND_2, M_{02}) is $n_1 + n_2$. Let n be the number of the max firing sequences in these two models [i.e., $n = max(n_1, n_2)$], and then their projected firing sequences can be found in $O(n^2)$. The calculation of trace consistency degree is in linear time with the calculation of projected firing sequences. Therefore, the time complexity of this algorithm is $O(n^2)$.

According to the DTC method, there are 10 traces in Figures 1A, B if data operations and guard functions are ignored. The traces in Figures 1A, B are $t_1t_2\{(t_3)(t_5t_6)\}t_8$, $t_1t_2\{(t_4)(t_5t_6)\}t_8$, $t_1t_2\{(t_3)(t_7)\}t_8$, and $t_1t_2\{(t_4)(t_7)\}t_8$, as shown in Figure 4A. Therefore, the trace consistency degree of these two models computed by the DTC method is 1.

However, there are 5 traces (resp. 4 traces) in Figure 1A (resp. Figure 1A) when data operations and guard functions are considered. The traces in Figure 1A are $t_1t_2\{(t_3)(t_5t_6)\}t_8$, $t_1t_2\{(t_4)(t_7)\}t_8$, and the traces in Figure 1B are $t_1t_2\{(t_3)(t_5)\}t_6t_8$ and $t_1t_2\{(t_4)(t_7)\}t_8$, as shown in Figures 4B, C. Therefore, the trace consistency degree of these two models is $TCDW^{\sim} = \frac{4+4}{5+4} \approx 0.889$.

4. Experiments

To evaluate the trace consistency of eight pieces of pseudo-code (i.e., Tables 2, 3), we use eight WFD-net systems to model them, as shown in Figures 5, 8. Although the method of DTC (Weidlich et al., 2011) can be used to measure the trace consistency degree, we want to address the following three problems.

RQ1: How do guard functions affect the traces in WFD-net systems?
RQ2: How do data dependencies affect the traces in WFD-net systems?
RQ3: Is our method more effective in measuring the trace consistency degree?

To answer RQ1-RQ3, we analyze the above-mentioned eight WFD-net systems. These systems include weak strict order relations, weak exclusiveness relations, and weak interleaving order relations. To compare our method with the DTC method, we analyze the trace consistency degree of these models with guard functions and data dependencies.

4.1. RQ1: The effect of guard functions on traces

As shown in Section 3, we know the guard functions may change the behavioral relations. Therefore, we discuss the traces in the general cases (e.g., Figure 5A) as follows:

(1) If there exists no data item d satisfying $d \in Rel(Gd(t_x)) \cap Rel(Gd(t_y))$, we have $(t_2 \parallel_{cd} t_4) \wedge (t_2 \parallel_{cd} t_5) \wedge (t_3 \parallel_{cd} t_4) \wedge (t_3 \parallel_{cd} t_5)$ in Figure 5A. The corresponding traces are illustrated in Figure 6A.
(2) If $\Gamma\left(Rel\left(Gd\left(t_x\right)\right)\right) = \Gamma(Rel\left(Gd\left(t_y\right)\right))$, then we have $(t_2 \parallel_{cd} t_4) \wedge (t_2 +_{cd} t_5) \wedge (t_3 +_{cd} t_4) \wedge (t_3 \parallel_{cd} t_5)$ in Figure 5A. The corresponding traces are illustrated in Figure 6B.
(3) If $\Gamma\left(Rel\left(Gd\left(t_x\right)\right)\right) = \Gamma(Rel\left(\neg Gd\left(t_y\right)\right))$, then we have $(t_2 +_{cd} t_4) \wedge (t_2 \parallel_{cd} t_5) \wedge (t_3 \parallel_{cd} t_4) \wedge (t_3 +_{cd} t_5)$ in Figure 5A. The corresponding traces are illustrated in Figure 6C.
(4) If $Rel\left(Gd\left(t_x\right)\right) \cap Rel\left(Gd\left(t_y\right)\right) = \{\mathfrak{D}\}$ and $\forall d \in \mathfrak{D}: [Rel\left(Gd\left(t_x\left(d\right)\right)\right) \cap \Gamma(Rel\left(Gd\left(t_x\left(d\right)\right)\right)) \neq \varnothing] \wedge [Rel\left(Gd\left(t_y\left(d\right)\right)\right) \cap \Gamma(Rel\left(Gd\left(t_y\left(d\right)\right)\right)) \neq \varnothing]$, we have $(t_2 \parallel_{cd} t_4) \wedge (t_2 +_{cd} t_5) \wedge (t_3 +_{cd} t_4) \wedge (t_3 +_{cd} t_5)$ in Figure 5A. The corresponding traces are illustrated in Figure 6D.
(5) If $Rel\left(Gd\left(t_x\right)\right) \cap Rel\left(\neg Gd\left(t_y\right)\right) = \{\mathfrak{D}\}$, and $\forall d \in \mathfrak{D}: [Rel\left(Gd\left(t_x\left(d\right)\right)\right) \cap \Gamma(Rel\left(Gd\left(t_x\left(d\right)\right)\right)) \neq \varnothing] \wedge [Rel\left(\neg Gd\left(t_y\left(d\right)\right)\right) \cap \Gamma(Rel\left(\neg Gd\left(t_y\left(d\right)\right)\right)) \neq \varnothing]$, we have

A	B	C
$t_1t_2\{(t_3)(t_5t_6)\}t_8$ $t_1t_2\{(t_4)(t_5t_6)\}t_8$ $t_1t_2\{(t_3)(t_7)\}t_8$ $t_1t_2\{(t_4)(t_7)\}t_8$	$t_1t_2\{(t_3)(t_5t_6)\}t_8$ $t_1t_2\{(t_4)(t_7)\}t_8$	$t_1t_2\{(t_3)(t_5)\}t_6t_8$ $t_1t_2\{(t_4)(t_7)\}t_8$

FIGURE 4
DTC (Weidlich et al., 2011) method: **(A)** shows the traces of Figures 1A, B. Our method: **(B)** reveals the traces of Figure 1A, **(C)** illustrates the traces of Figure 1B.

TABLE 2 Four kinds of pseudo-codes with guard functions.

1	2	3	4
$\{t_1\}$	$\{t_1\}$	$\{t_1\}$	$\{t_1\}$
Thread1()	If $(Gd(t_x))$	If $(Gd(t_x))$	If $(Gd(t_x))$
If $(Gd(t_x))$	Thread1()	Thread1()	Thread1()
$\{t_2\}$	$\{t_2\}$	$\{t_2\}$	$\{t_2\}$
Else	Thread2()	Thread2()	Thread2()
$\{t_3\}$ If $(Gd(t_y))$	$\{t_3\}$	$\{t_4\}$	$\{t_5\}$
End if	Else	Else	Else
Thread2()	Thread3()	Thread3()	Thread3()
If $(Gd(t_y))$	$\{t_4\}$	$\{t_3\}$	$\{t_3\}$
$\{t_4\}$	Thread4()	Thread4()	Thread4()
Else	$\{t_5\}$	$\{t_5\}$	$\{t_4\}$
$\{t_5\}$	End if	End if	End if
End if	$\{t_6\}$	$\{t_6\}$	$\{t_6\}$
$\{t_6\}$			

TABLE 3 Four kinds of pseudo-codes with data dependencies.

1	2
$\{t_1\}$	$\{t_1\}$
$\{t_2\}: int * d = new\ int,\ * d = 3$	Thread1()
$\{t_3\}: c = d, delete\ (d)\ , d = NULL$	$\{t_2\}: int * d = new\ int,\ * d = 3$
$\{t_4\}$	Thread2()
	$\{t_3\}: c = d, delete\ (d)\ , d = NULL$
	$\{t_4\}$
3	**4**
$\{t_x\}: int * d = new\ int,\ * d = 3$	$\{t_y\}: int * d = new\ int,\ * d = 3$
$\{t_1\}$	$\{t_1\}$
$\{t_2\}: int * d = new\ int,\ * d = 3$	Thread1()
$\{t_3\}: c = d, delete\ (d)\ , d = NULL$	$\{t_2\}: int * d = new\ int,\ * d = 3$
$\{t_4\}$	Thread2()
	$\{t_3\}: c = d, delete\ (d)\ , d = NULL$
	$\{t_4\}$

$(t_2 +_{cd} t_4) \wedge (t_2 \parallel_{cd} t_5) \wedge (t_3 +_{cd} t_4) \wedge (t_3 +_{cd} t_5)$ in Figure 5A. The corresponding traces are illustrated in Figure 6E.

(6) If $Rel\left(\neg Gd\left(t_x\right)\right) \cap Rel\left(Gd\left(t_y\right)\right) = \{\mathfrak{D}\}$, and $\forall d \in \mathfrak{D}: [Rel(\neg Gd\left(t_x\left(d\right)\right)) \cap \Gamma(Rel\left(\neg Gd\left(t_x\left(d\right)\right)\right)) \neq \varnothing] \wedge [Rel\left(Gd\left(t_y\left(d\right)\right)\right) \cap \Gamma(Rel\left(Gd\left(t_y\left(d\right)\right)\right)) \neq \varnothing]$, we have $(t_2 +_{cd} t_4) \wedge (t_2 +_{cd} t_5) \wedge (t_3 \parallel_{cd} t_4) \wedge (t_3 +_{cd} t_5)$ in Figure 5A. The corresponding traces are illustrated in Figure 6F.

(7) If $Rel\left(\neg Gd\left(t_x\right)\right) \cap Rel\left(\neg Gd\left(t_y\right)\right) = \{\mathfrak{D}\}$, and $\forall d \in \mathfrak{D}: [Rel(\neg Gd\left(t_x\left(d\right)\right)) \cap \Gamma(Rel\left(\neg Gd\left(t_x\left(d\right)\right)\right)) \neq \varnothing] \wedge [Rel\left(\neg Gd\left(t_y\left(d\right)\right)\right) \cap \Gamma(Rel\left(\neg Gd\left(t_y\left(d\right)\right)\right)) \neq \varnothing]$, we have $(t_2 +_{cd} t_4) \wedge (t_2 +_{cd} t_5) \wedge (t_3 +_{cd} t_4) \wedge (t_3 \parallel_{cd} t_5)$ in Figure 5A. The corresponding traces are illustrated in Figure 6G.

In our method, Figure 6H shows the traces of Figure 5B, Figure 6B shows the traces of Figure 5C, and Figure 6C shows the traces of Figure 5D. Therefore, the traces in a WFD-net system are more than one when their guard functions are uncertain, e.g., Figure 5A. Figure 7 shows the trace consistency degrees of models in Figures 5A–D.

4.2. RQ2: The effect of data dependencies on traces

In this subsection, we discuss the traces that may be affected by data dependencies, as shown in Figure 8. Based on Def. 7, the data dependencies in Figures 8A, C, D cannot change the traces. Although we have $d \lessdot MD$ in Figure 8B, the final place can be reached. The data dependencies in Figure 8B can change the traces if there is no locking mechanism. In a word, we have $t_2 \rightarrow_{cd} t_3$ in Figure 8B. Therefore, the corresponding traces of Figures 8A–D are shown in Figure 9. Figure 10 shows the trace consistency degrees of models in Figures 8A–D.

4.3. RQ3: Comparison between different trace consistency measurement methods

In this subsection, we present a comparison between different methods for trace consistency measurements. The

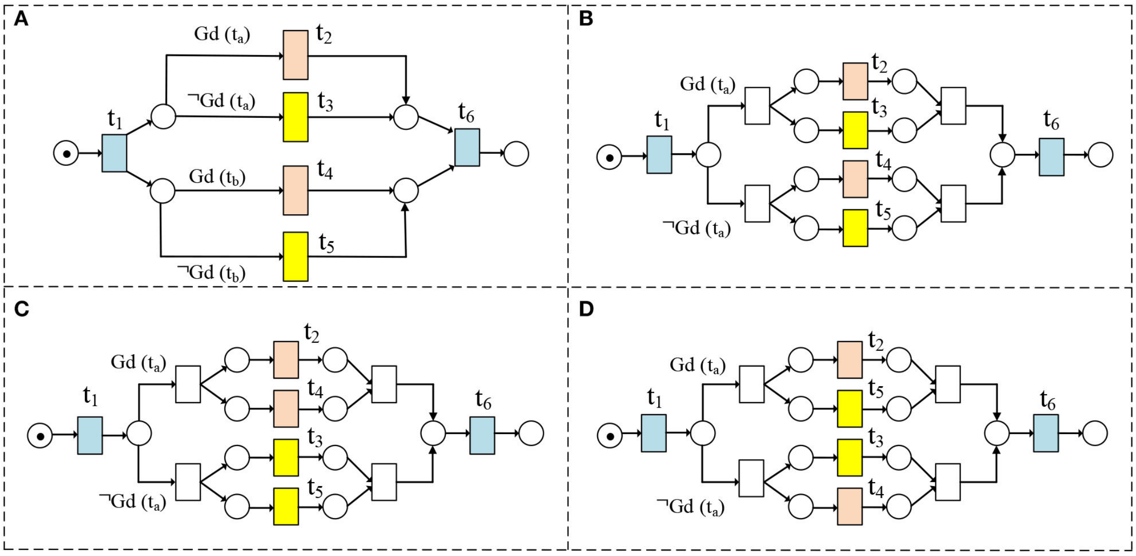

FIGURE 5
(A–D) are WFD-net models of Table 2.

$t_1\{(t_2)(t_4)\}t_6$ $t_1\{(t_2)(t_5)\}t_6$ $t_1\{(t_3)(t_4)\}t_6$ $t_1\{(t_3)(t_5)\}t_6$ (A)	$t_1\{(t_2)(t_4)\}t_6$ $t_1\{(t_3)(t_5)\}t_6$ (B)	$t_1\{(t_2)(t_5)\}t_6$ $t_1\{(t_3)(t_4)\}t_6$ (C)	$t_1\{(t_2)(t_4)\}t_6$ (D)
$t_1\{(t_2)(t_5)\}t_6$ (E)	$t_1\{(t_3)(t_4)\}t_6$ (F)	$t_1\{(t_3)(t_5)\}t_6$ (G)	$t_1\{(t_2)(t_3)\}t_6$ $t_1\{(t_4)(t_5)\}t_6$ (H)

FIGURE 6
DTC (Weidlich et al., 2011) method: **(A)** shows the traces of Figure 5A, **(H)** shows the traces of Figure 5B, **(B)** shows the traces of Figure 5C, and **(C)** shows the traces of Figure 5D. Our method: **(A–G)** shows the traces of Figure 5A, **(H)** shows the traces of Figure 5B, **(B)** shows the traces of Figure 5C, and **(C)** shows the traces of Figure 5D.

comparison includes two methods in total. The DTC method does not consider the trace consistency degree affected by guard functions and data dependencies of transitions. The results of our method (i.e., TCDW) take both sides into consideration, especially when the values of guard functions are uncertain.

In Figure 7, we can get 3 kinds of trace consistency degrees of Figures 5A, C as the values of guard functions are uncertain, i.e., $TCDW_1 = 0$, 0.667 or 1. Figures 7, 10 present the comparison results of the DTC method with the TCDW method on trace consistency measurement. Obviously, our result is more reasonable than the existing methods.

5. Related work

Consistency checking has become a critical technology in checking how a process model is similar to another. It is significant in the matching process models (Weidlich et al., 2012). Existing studies on the measurements of consistency degree can

<table>
<tr><th></th><th>Fig.5(A) and Fig.5(B)</th><th>Fig.5(A) and Fig.5(C)</th><th>Fig.5(A) and Fig.5(D)</th><th>Fig.5(B) and Fig.5(C)</th><th>Fig.5(B) and Fig.5(D)</th><th>Fig.5(C) and Fig.5(D)</th></tr>
<tr><td>DTC</td><td>0</td><td>0.667</td><td>0.667</td><td>0</td><td>0</td><td>0</td></tr>
<tr><td rowspan="7">TCDW</td><td>0</td><td>0.667</td><td>0.667</td><td rowspan="7">0</td><td rowspan="7">0</td><td rowspan="7">0</td></tr>
<tr><td>0</td><td>1</td><td>0</td></tr>
<tr><td>0</td><td>0</td><td>1</td></tr>
<tr><td>0</td><td>0.667</td><td>0</td></tr>
<tr><td>0</td><td>0</td><td>0.667</td></tr>
<tr><td>0</td><td>0</td><td>0.667</td></tr>
<tr><td>0</td><td>0.667</td><td>0</td></tr>
</table>

FIGURE 7
Trace consistency degree measurement results by taking DTC (Weidlich et al., 2011) and TCDW methods.

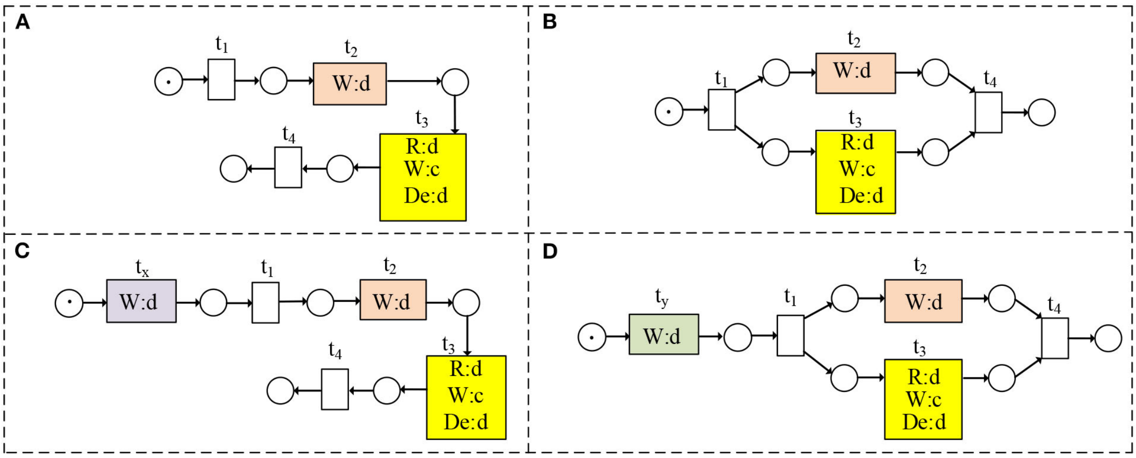

FIGURE 8
(A–D) are WFD-net models of Table 3.

be classified into two categories: with or without data. The following parts review some related work about the existing measuring methods.

5.1. Consistency degree of process models without data

Consistency measurement is very important in process alignment and process integration. It can be taken to detect the deviations between the process models and corresponding event logs. Recently, many researchers have focused on measuring the consistency degrees of process models from the angle of control flow.

Decker and Weske (2007) presented the standard for an optimum consistency relation regarding a given compatibility relation. They guaranteed that the implemented services' behaviors were compatible with others. Therefore, they checked the consistency degree of specifications and service implementations. After then, van der Aalst (2011) and van der Aalst et al. (2007) depicted the conformity relationship between event logs and process models. The consistency measurement techniques can be used to compare the modeled behavior in the initial process model with the observed behavior in event logs. The consistency guarantees there is no contradiction between the two models (Rull et al., 2008). Rozinat and van der Aalst (2008) proposed a method to check the consistency of an event log to a process model. This method is based on monitoring real behavior. Unfortunately, these studies cannot distinguish between slight inconsistencies and complete inconsistencies.

Based on existing research, Weidlich and Mendling (2010), Weidlich and van der Werf (2012), and Weidlich et al. (2011) solved this problem. They defined behavioral profile relations that can capture the behavioral restrictions of a model. Moreover, they put forward the trace consistency as the number of the same traces in two models to the number of traces with transitions in correspondences. Additionally, based on correspondence relations, they defined the profile consistency degree. They also illustrated that consistency-checking technology states how the real behavior is consistent with the observed. After then, Nagel et al. (2013) described process model quality constraints that can be taken to obtain consistency among the goals and the process models. van der Aa et al. (2015) used natural language processing techniques to recognize the inconsistencies between textual descriptions and relevant process models. Awadid and Nurcan (2016) reviewed the existing consistency measurement methods. One aim of their research is to master the inter-model consistency issue. Besides, Wang et al. (2018) proposed relation profile methods to analyze the consistency degrees of different process models. Although these consistency measurement methods have been proposed, they mostly rely on detecting the consistency decrease caused by the difference in process models. They have not taken the data dependencies or guard functions into consideration.

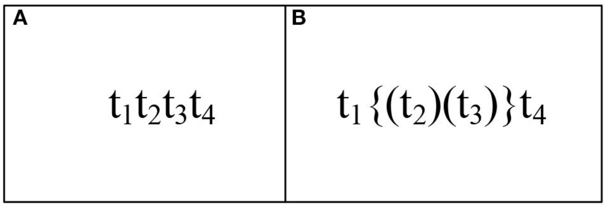

FIGURE 9
DTC (Weidlich et al., 2011) method: **(A)** reveals the traces of Figures 8A, C, **(B)** shows the traces of Figures 8B, D. Our method: **(A)** illustrates the traces of Figures 8A–C, **(B)** shows the traces of Figure 8D.

5.2. Consistency degree of process models with data

Generally speaking, the consistency degree of process models in workflow systems would be affected by the data. Song et al. (2013) and Song and Jacobsen (2015) proposed a way to calculate the consistency degrees between conceptual and executable process models. At the same time, they developed a tool named COCO to quantitatively calculate the consistency degree. After then, they measured the trace consistency degree between the two models on account of control dependence and data dependence and provided a new computing method to quantify the trace consistency (Song et al., 2021). Based on the program dependency graphs, Zhang et al. (2018) and Zhang et al. (2019) proposed a systematic method to check consistencies between different BPEL processes. They identified that the data dependencies can affect the consistency degrees of different models. They took advantage of activity constraint graphs to calculate the consistency degree of models with control flow and data-flow information. After taking their methods, we can distinguish different event constraints and check the consistency of conceptual and logical process models.

However, these studies are mainly based on analyzing the consistency of physical processes with read/write operations. They have not considered that the guard functions may affect the trace consistency degree. Usually, the transitions without correspondence relations but with data dependencies are ignored by them, too. However, all of them may affect the real trace consistency degree. By comparison, our measurement methods can solve these problems.

6. Conclusions and future work

The trace consistency measurement is important to service-based workflow net systems. This paper presents the behavioral profile relation in a WFD-net system, and provides a new way to calculate the trace consistency of two WFD-net systems. Compared with the DTC method, the advantages of our method are shown as

	Fig.8(A) and Fig.8(B)	Fig.8(A) and Fig.8(C)	Fig.8(A) and Fig.8(D)	Fig.8(B) and Fig.8(C)	Fig.8(B) and Fig.8(D)	Fig.8(C) and Fig.8(D)
DTC	0.667	1	0.667	0.667	1	0.667
TCDW	1	1	0.667	1	0.667	0.667

FIGURE 10
Trace consistency degree measurement results by taking DTC (Weidlich et al., 2011) and TCDW methods.

follows. Firstly, we illustrate that the guard functions may affect the traces in two WFD-net systems. Therefore, the trace consistency degree is non-uniqueness when the values of guard functions are uncertain. Secondly, we point out that the transitions with or without correspondence but with some data dependencies can also affect the traces and the real trace consistency degree. Some experiments on WFD-net systems above demonstrate that our method is more efficient in measuring trace consistency than the existing method.

This paper shows some directions for future research:

(1) Our methods need to be improved so as to measure the trace consistency between transitions in WFD-net systems extended with complex correspondence relations;
(2) Some measures should be taken to repair control/data-flow errors as they may affect the actual traces; and
(3) We need to develop a tool to help us calculate trace consistency automatically.

Data availability statement

The original contributions presented in the study are included in the article/supplementary material, further inquiries can be directed to the corresponding author.

Author contributions

Before writing this paper, FZ, DX, and GL have discussed the relevant details about trace consistency measurement. Afterwards, FZ wrote the paper. DX and GL revised the paper. All authors contributed to the article and approved the submitted version.

Funding

This work was partially supported by the National Nature Science Foundation of China under Grant 62002328, 62172299 and 62032019, the Youth Foundation of Anhui University of Technology under Grant QZ202215, and the Key Laboratory of Embedded System and Service Computing under Grant ESSCKF 2021-02.

Conflict of interest

The authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

Publisher's note

All claims expressed in this article are solely those of the authors and do not necessarily represent those of their affiliated organizations, or those of the publisher, the editors and the reviewers. Any product that may be evaluated in this article, or claim that may be made by its manufacturer, is not guaranteed or endorsed by the publisher.

References

Assy, N., Chan, N. N., and Gaaloul, W. (2015). An automated approach for assisting the design of configurable process models. *IEEE Trans. Serv. Comput.* 8, 874–888. doi: 10.1109/TSC.2015.2477815

Awadid, A., and Nurcan, S. (2016). A systematic liturature review of consistency among business process models. *BPMDS/EMMSAD 2016 LNBIP* 248, 175–195. doi: 10.1007/978-3-319-39429-9_12

Decker, G., and Weske, M. (2007). "Behavioral consistency for B2B process integration," in *19th International Conference on Advanced Information System Engineering, CAiSE 2007* (Trandheim: Springer), 81–95. doi: 10.1007/978-3-540-72988-4_7

Dijkman, R., Dumas, M., van Dongen, B., Kaarik, R., and Mendling, J. (2011). Similarity of business process models: metrics and evaluation. *Inf. Syst.* 36, 498–516. doi: 10.1016/j.is.2010.09.006

Fahland, D. (2008). *Translating UML2 Activity Diagrams to Petri Nets*. Berlin, Germany: Humboldt Universitat zu Berlin.

IEEE (1990). IEEE standard glossary of software engineering terminology. *Inst. Electrical Electron. Eng.* 42, 112–118.

Li, H., El-Amine, H. M., and Mohamed, H. (2014). "Merging several business process variants," in *The 26th Chinese Control and Decision Conference* (Changsha, China), 2, 5218–5223. doi: 10.1109/CCDC.2014.6853112

Li, Z. W., and Liu, G. Y. (2009). An iterative synthesis approach to petri net based on deadlock prevention policy for flexible manufacturing systems. *IEEE Trans. Syst. Man. Cybern. Syst.* 39, 362–371. doi: 10.1109/TSMCA.2009.2013446

Liu, G. J. (2014). Some complexity results for the soundness problem of workflow nets. *IEEE Trans. Serv. Comput.* 7, 322–328. doi: 10.1109/TSC.2013.36

Liu, G. J. (2022). *Petri Nets: Theoretical Models and Analysis Methods for Concurrent Systems*. Springer. doi: 10.1007/978-981-19-6309-4

Liu, G. J., Jiang, C. J., and Zhou, M. C. (2018). Time-soundness of time Petri nets modeling time-critical systems. *ACM Transac. Cyber Phys. Syst.* 2, 11. doi: 10.1145/3185502

Liu, G. J., Jiang, C. J., Zhou, M. C., and Xiong, P. C. (2013). Interactive Petri nets. *IEEE Trans. Syst. Man Cybernet. Syst.* 43, 291–302. doi: 10.1109/TSMCA.2012.2204741

Liu, G. J., Zhou, M. C., and Jiang, C. J. (2017). Petri net modelling and collaborativeness for parallel processes with resource sharing and message passing. *ACM Trans. Embedded Comput. Syst.* 16, 1–20. doi: 10.1145/2810001

Liu, G. Y., Li, P., Li, Z. W., and Wu, N. Q. (2019). Robust deadlock control for automated manufacturing systems with unreliable resources based on Petri net reachability graphs. *IEEE Trans. Syst. Man. Cybern. Syst.* 49, 7, 1371–1385. doi: 10.1109/TSMC.2018.2815618

Meda, H. S., Sen, A. K., and Bagchi, A. (2010). On detecting data flow errors in workflows. *ACM J. Data Inf. Qual.* 2, 1–31. doi: 10.1145/1805286.1805290

Nagel, B., Gerth, C., Engels, G., and Post, J. (2013). "Ensuring consistency among business goals and business process models," in *IEEE Enterp. Distrib. Object Comput* (Washington, DC, USA), 17–26. doi: 10.1109/EDOC.2013.12

Rozinat, A., and van der Aalst, W. M. P. (2008). Conformance checking of processes based on monitoring real behavior. *Inf. Syst.* 33, 64–95. doi: 10.1016/j.is.2007.07.001

Rull, G., Farre, C., Teniente, E., and Urpf, T. (2008). Validation of mappings between schemas. *Data Knowl. Eng.* 66, 414–437. doi: 10.1016/j.datak.2008.04.009

Sidorova, N., Stahl, C., and Trcka, N. (2011). Soundness verification for conceptual workflow nets with data: early detection of errors with the most precision possible. *Inf. Syst.* 36, 1026–1043. doi: 10.1016/j.is.2011.04.004

Song, W., and Jacobsen, H. A. (2015). Static and dynamic process change. *IEEE Trans. Serv. Comput.* 11, 215–231. doi: 10.1109/TSC.2016.2536025

Song, W., Jacobsen, H. A., Cheung, S. C., Liu, H. Y., and Ma, X. X. (2018). Workflow refactoring for maximizing concurrency and block-structuredness. *IEEE Trans. Serv. Comput*., 14, 1224–1237. doi: 10.1109/TSC.2018.2867593

Song, W., Jacobsen, H. A., Ye, C. Y., and Ma, X. X. (2016). Process discovery from dependence-complete event logs. *IEEE Trans. Serv. Comput*. 9, 714–727. doi: 10.1109/TSC.2015.2426181

Song, W., Jacobsen, H. A., Zhang, C. Z., and Ma, X. X. (2021). Dependence-based data-aware process conformance checking. *IEEE Trans. Serv. Comput*. 14, 654–667. doi: 10.1109/TSC.2018.2821685

Song, W., Zhang, W. J., Zhang, G. X., Ding, J. H., and Zhang, X. W. (2013). "Quantifying consistency between conceptual and executable business processes," in *2013 IEEE International Conference on Services Computing* (Santa Clara, CA: IEEE), 9–16. doi: 10.1109/SCC.2013.47

Trcka, N., van der Aalst, W. M. P., and Sidorova, N. (2009). *Data-Flow Anti-Patterns: Discovering Data-Flow Errors in Workflows*. Berlin, Heidelberg: Springer, 425–439. doi: 10.1007/978-3-642-02144-2_34

van der Aa, H., Leopold, H., and Reijers, H. A. (2015). *Detecting Inconsistencies Between Process Models and Textual Descriptions*.Cham: Springer, 90–105. doi: 10.1007/978-3-319-23063-4_6

van der Aalst, W. M. P. (2011). Service mining: using process mining to discover, check, and improve service behavior. *IEEE Trans. Services Comput*. 6, 1–13. doi: 10.1109/TSC.2012.25

van der Aalst, W. M. P., Dumas, M., Gottschalk, F., ter Hofstede, A. H. M., La Rosa, M., and Mendling, J. (2010). Preserving correctness during business process model configuration. *Formal Aspects Comput*. 22, 459–482. doi: 10.1007/s00165-009-0112-0

van der Aalst, W. M. P., Reijers, H. A., and Werjters, A. J. M. (2007). Business process mining: an industrial application. *Inf. Syst*. 32, 713–732. doi: 10.1016/j.is.2006.05.003

Wang, M. M., Ding, Z. J., Liu, G. J., Jiang, C. J., and Zhou, M. C. (2018). Measurement and computation of profile similarity of workflow nets based on behavioral relation matrix. *IEEE Trans. Sys., Man Cybern. Syst*. 50, 1–18. doi: 10.1109/TSMC.2018.2852652

Weidlich, M., Dijkman, R., and Weske, M. (2012). Behavior equivalence and compatibility of business process models with complex correspondences. *Comput. J*. 55, 1398–1418. doi: 10.1093/comjnl/bxs014

Weidlich, M., Dijkman, R., and Weske, M. (2014). "Efficient analysis of BPEL 2.0 processes using p-calculus," in *Proceedings of IEEE 30th International Conference on Data Engineering* (Chicago, IL: IEEE), 376–387.

Weidlich, M., and Mendling, J. (2010). Perceived consistency between process models. *Inf. Syst*. 37, 80–98. doi: 10.1016/j.is.2010.12.004

Weidlich, M., Mendling, J., and Weske, M. (2011). Efficient consistency measurement based on behavioral profiles of process models. *IEEE Trans. Softw. Eng*. 37, 410–429. doi: 10.1109/TSE.2010.96

Weidlich, M., and van der Werf, J. J. (2012). On profiles and footprints relational semantics for Petri nets. *Applic. Theor. Petri Nets*. 7347, 148–167. doi: 10.1007/978-3-642-31131-4_9

Xiang, D. M., Liu, G. J., Yan, C. G., and Jiang, C. J. (2021). A guard-driven analysis approach of workflow net with data. *IEEE Trans. Services Comput*. 14, 1650–1661. doi: 10.1109/TSC.2019.2899086

Zhang, X. W., Song, W., Wang, J. J., and Xing, J. C., and Zhou, Q. Z. (2019). Measuring business process consistency across different abstraction levels. *IEEE Trans. Netw. Serv. Manag.* 16, 294–307. doi: 10.1109/TNSM.2018.2883362

Zhang, X. W., Wang, J. J., Xing, J. C., Song, W., and Yang, Q. L. (2018). Measuring data-aware process consistency based on activity constraint graphs. *IEEE Access* 6, 21005–21019. doi: 10.1109/ACCESS.2018.2795701

Zhang, Z. L., Liu, G. Y., Barkaoui, K., and Li, Z. W. (2022). Adaptive deadlock control for a class of Petri nets with unreliable resources. *IEEE Trans. Syst. Man. Cybern. Syst*. 52, 3113–3125. doi: 10.1109/TSMC.2021.3062469

Zhao, F., Xiang, D. M., Liu, G. J., and Jiang, C. J (2022a). A new method for measuring the behavioral consistency degree of WF-net systems. *IEEE Trans. Comput. Social Syst*. 9, 480–493. doi: 10.1109/TCSS.2021.3099475

Zhao, F., Xiang, D. M., Liu, G. J., Jiang, C. J., and Zhu, H. H. (2022b). Detecting and repairing data-flow errors in WFD-net systems. *Comput. Modeling Eng. Sci*. 131, 1337–1363. doi: 10.32604/cmes.2022.018872

Zhu, X., Zhu, G., van den Broucke, S., and Recker, J. (2014). "On merging business process management and geographic information systems: modeling and execution of ecological concerns in processes," in *Communications in Computer and Information Science, vol 482* (Berlin, Heidelberg: Springer), 486–496. doi: 10.1007/978-3-662-45737-5_48