symmetry

MDPI

Article

Consistency Analysis of Collaborative Process Data Change Based on a Rule-Driven Method

Qianqian Wang [1] and Chifeng Shao [2,*]

[1] College of Information & Network Engineering, Anhui Science and Technology University, Bengbu 233000, China; wangqq@ahstu.edu.cn

[2] School of Mathematics and Big Data, Anhui University of Science and Technology, Huainan 232001, China

* Correspondence: fc_shao@126.com

Abstract: In business process management, business process change analysis is the key link to ensure the flexibility and adaptability of the system. The existing methods mostly focus on the change analysis of a single business process from the perspective of control flow, ignoring the influence of data changes on collaborative processes with information interaction. In order to compensate for this deficiency, this paper proposes a rule-driven consistency analysis method for data changes in collaborative processes. Firstly, it analyzes the influence of data changes on other elements (such as activities, data, roles, and guards) in collaborative processes, and gives the definition of data influence. Secondly, the optimal alignment technology is used to explore how data changes interfere with the expected behavior of deviation activities, and decision rules are integrated into the Petri net model to accurately evaluate and screen out the effective expected behavior that conforms to business logic and established rules. Finally, the initial optimal alignment is repaired according to the screened effective expected behavior, and the consistency of business processes is recalculated. The experimental results show that the introduced rule constraint mechanism can effectively avoid the misjudgment of abnormal behavior. Compared with the traditional method, the average accuracy, recall rate, and F1-score of effective expected behavior are improved by 4%, 4.7%, and 4.3%, respectively. In addition, the repaired optimal alignment significantly enhances the system's ability to respond quickly and self-adjust to data changes, providing a strong support for the intelligent and automated transformation of business process management.

Keywords: data change; decision rules; consistency; collaborative process

check for updates

Citation: Wang, Q.; Shao, C. Consistency Analysis of Collaborative Process Data Change Based on a Rule-Driven Method. *Symmetry* **2024**, *16*, 1233. https://doi.org/10.3390/sym16091233

Academic Editor: Antonio Palacios

Received: 20 August 2024
Revised: 9 September 2024
Accepted: 15 September 2024
Published: 20 September 2024

1. Introduction

Data are an important component of business processes [1,2]. With the variety of business requirements, business processes become complicated, especially collaborative processes with information interaction. Once a certain data point changes, a series of changes may occur in related processes, and the actual execution behavior of the process will deviate from the predefined behavior of the process. Therefore, it is very necessary to study data changes in collaborative processes.

At present, the research on change in the process is mainly divided into three aspects: change mining, change domain analysis, and change propagation. Change mining is an important branch of process mining, and it is the cross-integration of process mining and adaptive process management, aiming at discovering hidden changes in the operating system and contributing to process repair and optimization [3,4]. Günther et al. [5] proposed a method of mining change logs in an adaptive process management system, taking the mined changes as the basis of process improvement. Fang et al. [6] studied the use of incomplete logs and joint log relationships to mining change operations in logs when the model is unknown. Sun et al. [7] studied a behavior change mining method for complete logs with hidden transitions based on Fang et al. [6], and applied it to oil spill accidents to explore the dynamic evolution process of disaster chains in different scenarios. Fang et al. [8] proposed a log-induced change mining method and combined it with structural causal relationships to locate potential faults in the system. Hmami et al. [9,10] compared the merged and filtered mutation event logs with the mutation files to mine the change logs, and then used the mining results as the basis for the recommendation system.

Change region analysis is the analysis of differences between similar models. These differences are the unique characteristics and advantages of each model, which can provide a basis for process optimization. Weidlich et al. [11] first gives the change in the process model, and determines the change region in another model by using the behavior relationship in the behavior profile. This method can cope with the changes in model pairs unrelated to the hierarchical structure and can show the inconsistency of behavior. The disadvantage is that the degree of inconsistency may be increased by wrongly identifying the corresponding relationship. Zhao et al. [12] analyzed the suspicious change region of the target model and the source model from the perspective of behavior profile, and obtained the minimum change region of the target model by using dynamic slicing technology. Fang et al. [13] analyzed the change region propagation from two aspects, namely projection inheritance and protocol inheritance, and finally found the same change region of the target model. Fang et al. [14] dynamically analyzed the changeable region module of business process, and obtained the accurate changeable region module through T-variable. Zhao et al. [15] analyzed the change region by establishing a fusion process with the data flow and control flow, but did not consider the impact of data changes.

The essence of change propagation is to ensure the consistency of the same process at different levels, and to maintain the compatibility of the behavior and structure with related processes. Mahfouz et al. [16] introduced the concept of propagation, but how to spread it is not clear. Weidlich et al. [17] analyzed the behavioral relationships between nodes and identified the change areas caused by changes based on the structural characteristics of the model. At the same time, changes can be propagated to other similar processes, but this method only considers the control flow of the process and does not consider data attributes, which is not comprehensive enough. Kolb et al. [18] used a set of update operations to update the user view, propagated their changes to the underlying process model, and provided migration rules to ensure the consistency and compatibility of related processes. Fdhila et al. [19] and Dahman et al. [20] propagated the changes in centralized process model to its divided subprocesses based on the basic change pattern. Wootton et al. [21] pointed out that in processes where multiple process partners collaborate together, it may be more complicated to deal with changes, and changes in one process may have an impact on other process partners. Wang et al. [22] analyzed the influence of 10 change patterns of business processes on the basis of services, and applied these influences to the change propagation of services and processes. Dam et al. [23] predicted the impact of future changes by mining the version history in the business process repository. Experiments show that this method is more accurate than the analysis based on basic dependencies.

Based on the above methods, it is found that in the current business process management research, the analysis of process changes focuses on the dynamic adjustment and optimization of the control flow, ignoring the impact of data changes on the process. Data play an important role in the process, and the existence of data helps to describe the business process more comprehensively and to determine the accuracy of model execution. At present, the research on data is mainly in the fields of data flow error detection [24], model correctness verification [25], process consistency research [26], etc. But the analysis on how data changes directly and specifically affect the business process execution path and its results is still insufficient. Especially in the complex collaborative process environment, the sudden change in data may not only lead to the execution of a single process deviating from the preset model, but also aggravate this inconsistency due to the data dependence between multiple processes, which makes the stability and predictability of the overall business process face challenges. It is worth noting that although some data changes seem to be inconsistent with the expectations of the process, they may actually contain reasonable business logic or requirement changes. Therefore, how to effectively identify and repair such "reasonable deviations" to restore or optimize the execution behavior of the process has become an urgent problem to be solved.

Although the existing studies, such as references [27,28], have preliminarily explored the impact of data changes on process behavior and the corresponding adaptive mechanism, there is still a problem of insufficient accuracy in capturing the adaptive behavior of response deviation, because for data changes, we should not only consider how the data changes affect process behavior, but also analyze whether the data changes conform to the decision rules of process operation. In addition, the consistency of the data changes they studied was analyzed in a single process, which cannot solve the consistency analysis of collaborative processes with data dependence. Therefore, this paper proposes a rule-driven consistency analysis method for collaborative process data changes. The main contributions of this paper are as follows:

(1) Integrating the data information involved in the collaboration process into a decision analysis table, and establishing a decision Petri net model by combining it with a Petri net model to achieve an accurate description of the relationship between data changes and process behavior;
(2) A rule-driven effective expected behavior retrieval method is proposed, which obtains the expected behavior of deviation activities through optimal alignment, and verifies the effectiveness of expected behavior using decision rules, improving the accuracy of effective expected behavior and reducing the false negative rate;
(3) A method of repairing the alignment is proposed, and the consistency of business processes is improved by repairing the initial optimal alignment.

2. Motivating Example and Preliminaries

2.1. Motivating Example

Data are crucial for the operation of the processes. If we only consider the control flow of the process and ignore the data, it may lead to premature termination, incorrect route selection, deadlocks, and so on. Figure 1 shows a seller's sales process. Figure 1 is a real-life process design about the seller's sales process, which is independent of event logs. If there are real-life logs, they can be mined using mining tools. The common mining algorithms include the α algorithm [29], heuristic mining algorithm [30], genetic process mining [31], inductive mining [32], etc. The process begins with receiving the order and then checking the goods. There are three options, namely direct rejection, insufficient goods, and sufficient goods, and subsequent activities include sending order information, checking order information, checking the fee, checking payments, and delivery.

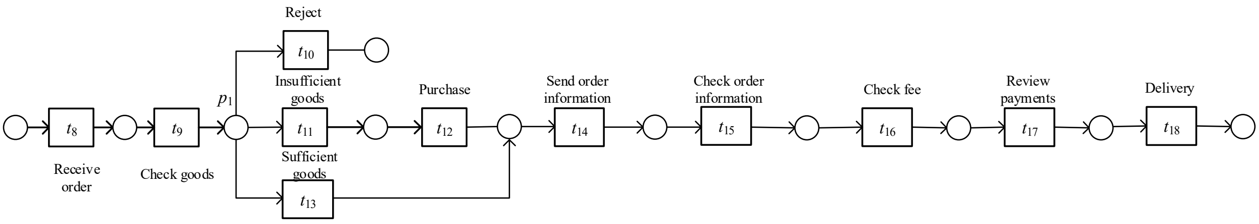


Figure 1. Petri net model.

In Figure 1, if the data are ignored and only the basic process behavior is considered, the seller may directly reject the order after checking the goods, and the process ends. It is also possible to directly load goods without purchasing, resulting in an inconsistency between the actual implementation of the later process and the behavior of the process. The three execution traces are as follows:

$$\sigma_1 = \langle t_8, t_9, t_{13}, t_{14}, t_{15}, t_{16}, t_9, t_{13}, t_{14}, t_{16}, t_{17}, t_{18} \rangle$$

$$\sigma_2 = \langle t_8, t_9, t_{13}, t_{14}, t_{15}, t_{16}, t_9, t_{11}, t_{12}, t_{14}, t_{16}, t_{17}, t_{18} \rangle$$

$$\sigma_3 = \langle t_8, t_9, t_{13}, t_{14}, t_{15}, t_{16}, t_9, t_{11}, t_{12}, t_{16}, t_{17}, t_{18} \rangle$$

It is obvious that none of these three traces meet the process model, as they all generate deviations after the activity check fee, which is also a normal phenomenon in real life. Because the requirements are constantly changing, the data in the business process may change, and the change in these data may cause a series of changes in system behavior. Especially in the collaborative process, the change in data will not only lead to changes within the process, but also leads to changes in the partner's process. Figure 2 is a collaborative process of commodity wholesale, involving three processes: buyer, seller, and logistics, in which the dotted line part represents some data interaction information. There is an unexpected situation where the seller has already reviewed the payment, and the buyer suddenly increases the quantity of goods. The seller finds that the goods are insufficient, which requires repurchasing. The freight may increase, and there may be data change propagation in the collaboration process. This change propagation is reasonable in practice, so the trace offset process may also be reasonable.

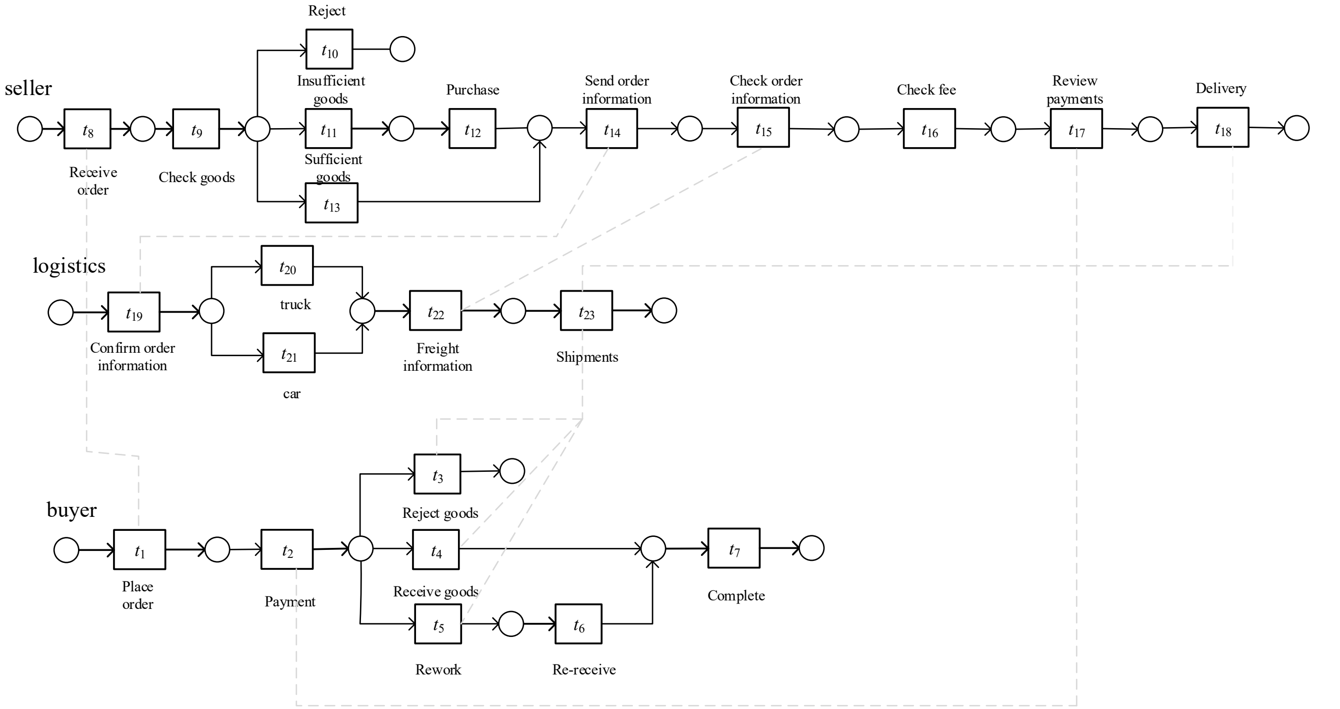


Figure 2. Collaboration process.

Example 1. *Assuming that the traces σ_2 and σ_3 are traces generated by the buyer's demand for goods, which changed after the seller checked the fees, and the goods are insufficient after the demand increased, it is necessary to purchase again, and the freight may change; σ_1 is produced because the quantity is sufficient after the demand is reduced, and the freight may also change. According to the method proposed by Mannhardt et al. [33], the consistency between the three traces and the process is calculated, respectively, and the fitness values of σ_1, σ_2, and σ_3 are 0.67, 0.62, and 0.67, respectively. It is found that the consistency of the three traces is not very high, and the fitness of σ_1 and σ_3 is the same. However, if some deviations may be reasonable considering the actual situation, the fitness of σ_1 and σ_3 is not necessarily the same, and it remains to be determined which of the three traces has the higher fitness. Take traces σ_2 and σ_3 as examples. Both of them are insufficient goods, and, in principle, they should be repurchased. However, if they do not meet the specified rules of data, they may be directly rejected without subsequent execution, resulting in a greater difference. Therefore, not only data but also data rules should be considered in the consistency of an alignment-based trace and process.*

2.2. Preliminaries

Definition 1 (Labeled Petri net [34]). *A 5-tuple $N = (P, T, F, \sum, \lambda)$ that satisfies the following conditions is called a labeled Petri net:*

(1) $P \cup T \neq \varnothing$;
(2) $P \cap T = \varnothing$;
(3) $F \subseteq (P \times T) \cup (T \times P)$;
(4) Σ *is the set of active labels for transitions;*
(5) $\lambda : T \to \sum$ *is a function of assigning labels to transitions;*

Where P is the place set, T is the transition set, and F is the flow relation.

Definition 2 (Event [35]). *Let ε be the event space, which is the set of all possible event identifiers, and AT is the set of all event attributes. For any event e ∈ ε , e = (case id, event id, activity, timestamp, attributes) is a tuple. case id, event id, activity ∈ T, timestamp, and attributes ∈ AT represent the process case identifier, event identifier, activity name, timestamp, and a set of other attributes of the event, respectively, and other attributes of different events in the trace may not be the same.*

Definition 3 (trace, event log [36]). *Trace* $\sigma = \langle e_1, e_2, e_3, \cdots, e_n \rangle$ *is a sequence of length n, where* $e \in \varepsilon^*$, ε^* *represents the set of all sequences on ε,* $\sigma(i)$ *represents the i-th event,* π_{at} *represents the value of the obtained event attribute at ∈ AT,* $1 \leq i < j \leq n$, $n = |\sigma|$, *and the event log* $L \in \mathrm{P}(\varepsilon^*)$ *is the set of traces,* P *represents a power set.*

3. Consistency Analysis of Data Changes Based on a Rule-Driven Method

3.1. Data Change Impact Analysis

Business processes often face various uncertain factors during execution, among which data change is one of the key factors affecting the accuracy of process execution. Especially in the collaborative process, the relationship between data sharing and dependence is more complicated, and data changes may lead to process behavior deviating from expectations, thus affecting the efficiency and effectiveness of the whole business process. Therefore, it is of great theoretical and practical significance to study the impact of data changes.

When the process is running, activities are executed according to the order determined in the process, and activities can read or write data; therefore, unexpected changes in data values may have an impact on activities, data, roles, and gateways in the process. The impact of data on activities is as follows: data affects the creation and execution of activities; when placing an order in the buyer's process, an order will be created, that is, the data item will be used as the input of the activity to create the output; in the seller's process, different activities are performed by different choices of goods quantity. The impact of data on data is as follows: the change in one datum causes another datum to change or remain unchanged; increase the purchase quantity of goods, and the freight may increase or remain the same. The impact of data on roles is as follows: an activity itself is executed by role A, and after the quantity changes, it may be executed by role B; with the change in the goods purchase, the delivery mode in the logistics process may be changed from a truck to a car. The impact of data on the gateway is as follows: a datum itself meets rule 1, and, after the change, rule 2 is true; if the goods qualification rate is between (0.85, 1) and becomes 1 after the rework, the goods will be received directly.

Definition 4 (Data impact). *Given a data element d and a trace σ, the active set represented by events in the trace σ and affected (directly or indirectly) by the value of d is called the data impact of d in σ, and is marked as DI (d, σ).*

For example, in Figure 2, the buyer's goods quantity suddenly decreases, and the seller finds that the goods quantity is sufficient based on the received order and does not need to repurchase. The corresponding data impact in part traces $\sigma = \langle t_8, t_9, t_{11}, t_{12} \rangle$ is $DI(goods\ quantity, \sigma) = \{t_{11}, t_{12}\}$.

3.2. Analysis of Effective Expected Behavior

During the process execution, there are various reasons that can cause changes in business processes, such as the implementation of new regulations and the emergence of new market demands [37], and so on. Once the data involved in the process change, the execution of activities may deviate from the actual model, and in the optimal alignment of process and log, it may be manifested as the movement of the log or model. For the activities after deviation, these are mainly a series of activities to make up for these deviations, which can be regarded as the expected behavior of deviation activities. In the optimal alignment, if the model moves, it means that the activity that should have occurred in the model did not occur, so the expected behavior is the activity that should have occurred but did not occur. If the log moves, it means that activities that should have occurred in the current location log have occurred. Assuming that the reason for this deviation is caused by changes in data related to previous execution activities, its expected behavior is the activity that may be affected by the data changes. If synchronous move occurs, that is, there is no deviation, then the expected behavior is recorded as an empty set. However, these expected behaviors may not be effective expected behaviors, because it remains to be determined whether the data context rules are met after some data changes. Therefore, in order to analyze the effective expected behavior based on optimal alignment under the impact of data, decision rules are introduced, which are represented as decision analysis tables in the process model.

Definition 5 (Alignment [38])**.** *Let the alignment of net N and trace σ be a sequence pair in the form of (e_i, t_i), where $e_i \in \sigma \cup \{\gg\}$, $t_i \in T \cup \{\gg\}$, if $e_i \neq \gg, t_i \neq \gg$, then $e_{i.activity} = t_i$.*

The three legal moves in alignment can be defined as follows:

(1) If $e_i \neq \gg, t_i \neq \gg$, (e_i, t_i) is a synchronous move;
(2) If $e_i \neq \gg, t_i = \gg$, (e_i, t_i) is a model move;
(3) If $e_i = \gg, t_i \neq \gg$, (e_i, t_i) is a log move.

Definition 6 (Cost function and optimal alignment [33])**.** *Let σ and N be a trace and a Petri net, respectively. Assuming Λ as the set of all legal alignment moves, a cost function c assigns a non-negative cost to each legal move: $\Lambda \to \mathbb{R}_0^+$. The cost of an alignment γ between σ and N is computed as the sum of the cost of all constituent moves: $C(\gamma) = \sum_{(e_i,t_i)\in\gamma} c(e_i, t_i)$. Alignment γ is an optimal alignment if, for any complete alignment γ' of N and σ, $c(\gamma) \leq c(\gamma')$.*

Definition 7 (Decision place)**.** *Let N and $p \in P$ be a Petri net and a place, respectively. If it is satisfied $|p^{\cdot}| > 1$, $p^{\cdot}$ represents the successor of p, and p is called a decision place.*

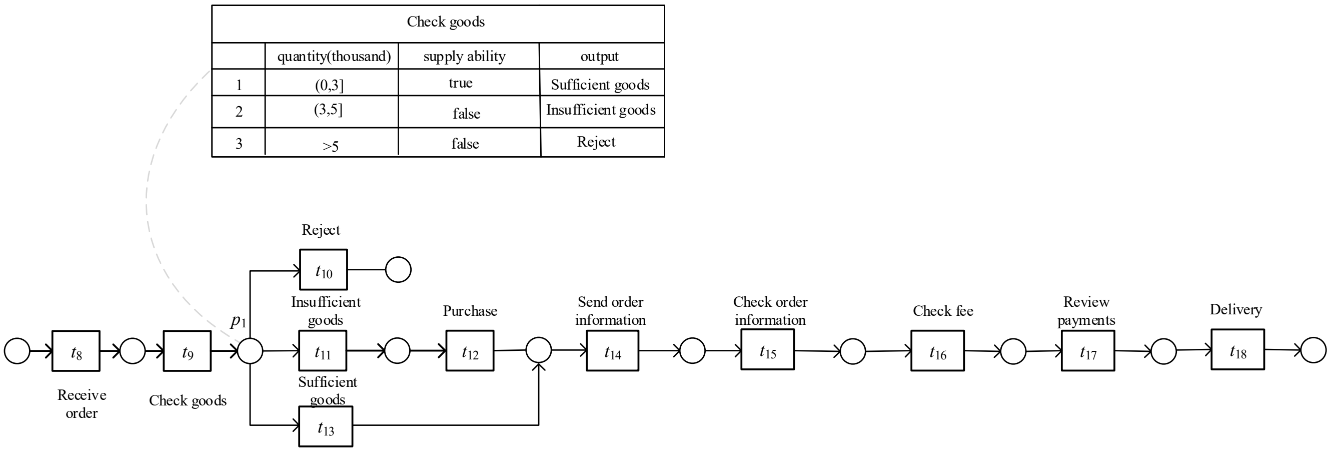
Definition 8 (Decision rule)**.** *Gives a set of attributes $Attr = \{at_1, at_2, \cdots, at_k\}, k \in N^*$, and decision rule r is a mapping function: $r(Attr) = c : \begin{array}{c}(at_1\ op\ q_1,\ at_1\ op\ q_2), \cdots,\\ (at_w\ op\ q_{2w-1},\ at_w\ op\ q_{2w}) \to c_l, 1 \leq w \leq k'\end{array}$ where attribute $at_1, at_2, \cdots, at_k$ is the input of decision; op is a comparison predicate; $q_1, q_2, \cdots, q_{2w}$ is a constant; c_l is the output of the decision, $c_l \in C_{p^{\cdot}}$, $h = |p^{\cdot}|, h, l \in N^*, 1 \leq l \leq h$, and $C_{p^{\cdot}} = \{c|\lambda(t), t \in p^{\cdot}\}$ is the activity name set of the successor of the decision place.*

For example, in Figure 3, the rule formed by a data dependency behind the decision place p_1 is as follows: $quantity \in (3,5] \to$ Sufficient goods.

Definition 9 (Decision analysis table)**.** *The decision analysis table $dt = (Name, I; O; R)$ is a tabular form, Name is the name of the table, I contains the attributes of the variables used for decision-making in the process model, and O is a finite non-empty set of outputs, where R is the set of decision rules r.*

It should be noted that the input–output set of the decision table is not fixed and unchanging. In this paper, only the output is used for output O, which indicates the name of the activity to occur, and the other three are not involved. For example, the decision analysis table involved in Figure 1 is shown in Figure 3.

Definition 10 (Decision Petri net)**.** *Let $DN = (N, V, U, DT)$ be a decision Petri net, where N is a label Petri net and V is a set of variables; U is a function to determine the possible value assigned to each variable, $U(v) = DO_v$, $v \in V$, and DO_v is the domain of v; DT is the set of decision analysis tables dt, $dt \in DT$; flow relation $F \subseteq (P \times T) \cup (T \times P) \cup (DT \times DP)$, and DP is the set of decision place p, $DP = \{p | p \in P, |p^{\cdot}| > 1\}$.*



Check goods

	quantity(thousand)	supply ability	output
1	(0,3]	true	Sufficient goods
2	(3,5]	false	Insufficient goods
3	>5	false	Reject

Figure 3. Decision Petri net.

In Algorithm 1, lines 1–2 initialize the expected behavior and valid expected behavior as empty sets, and then obtain the optimal alignment between the trace and the model DN. Lines 3–19 analyze the occurrence of expected behavior after synchronous move, log move, and model move, respectively, and when a synchronous move occurs, the expected behaviors remain an empty set. When a model move occurs, the expected behaviors are expanded by adding activity a; when a log move occurs, the algorithm first analyzes the data items associated with a, and then considers the impact set of those data items, which is the set of activities that should be added to the expected behaviors. Finally, Lines 20–24 evaluate whether the expected behaviors satisfy the decision rules to determine their validity. If an expected behavior does not meet the rule, it is considered anomalous and is removed from the effective expected behaviors set.

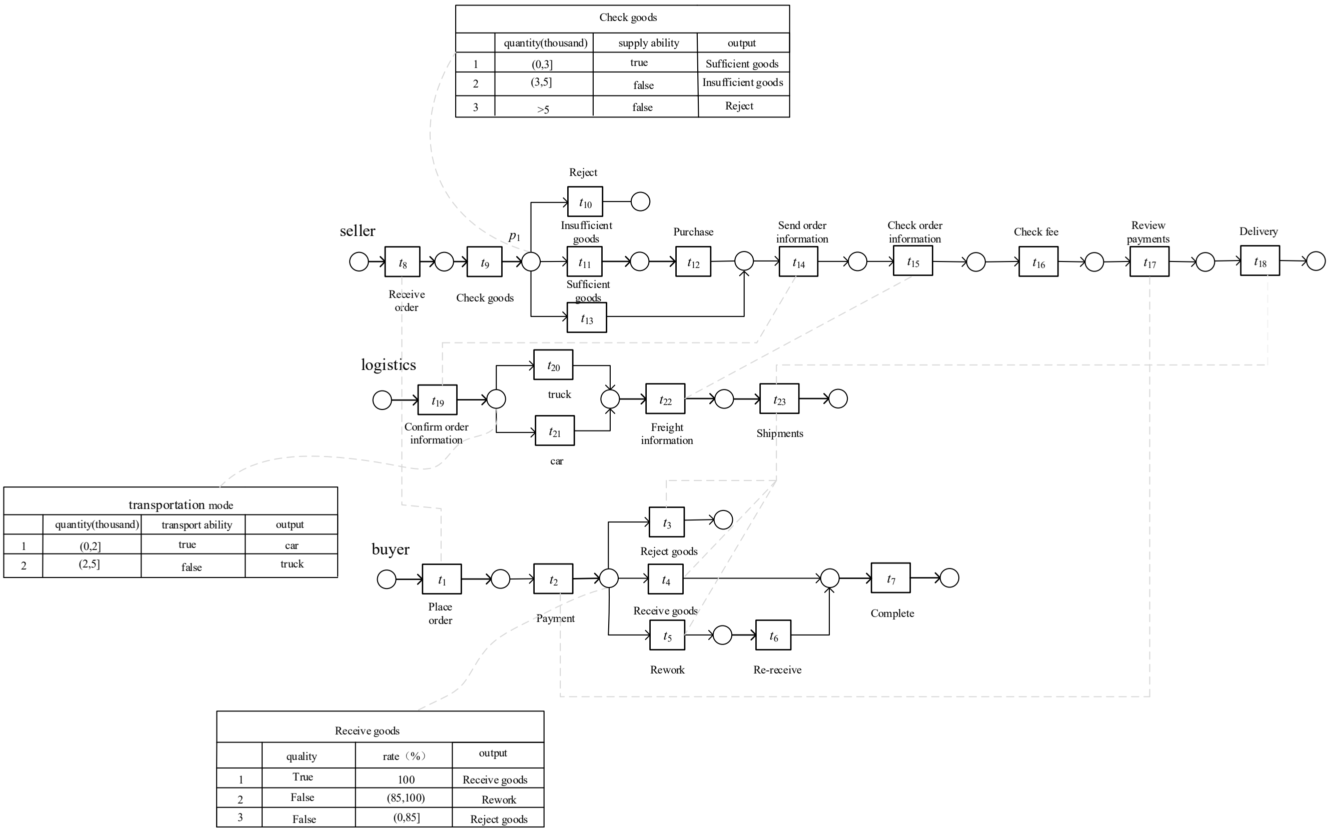
Algorithm 1. Rule-driven effective expected behavior retrieval method

```
Input: trace σ, decision Petri net DN
Output: effective expected behavior $\Theta'$
1.  $\Theta = \varnothing, \Theta' = \varnothing$
2.  $\gamma \leftarrow ObtainOptimalalignment(\sigma, N)$
3.  for each move $(e, a) \in \gamma$
4.      if e = $a$ then
5.          $\Theta \leftarrow \Theta$
6.      else if e =>> then
7.          $\Theta \leftarrow \Theta \cup \{a\}$
8.      else if $a$=>> then
9.          $Da \leftarrow Obtain\mathrm{Rel}atedDataItems(a)$
10.         if $Da = \varnothing$ then
11.             $\Theta \leftarrow \Theta$
12.         else if $Da \neq \varnothing$ then
13.             for each $d \in Da$ do
14.                 $DI \leftarrow Obtain\mathrm{Rel}atedActivity(d)$
```

Algorithm 1. *Cont.*

```
                $\Theta \leftarrow \Theta \cup DI$
            end for
        end if
    end if
end for
for each activity $\theta \in \Theta$ do
    if $r(\theta.Attr) = c$ is false then
        $\Theta' \leftarrow \Theta - \{\theta\}$
    end if
end for
return $\Theta'$
```

For example, consider a collaborative process involving buyers, sellers, and logistics with a decision analysis table, as shown in Figure 4. We will now analyze the specific execution of three traces in a motivating example. Under normal circumstances, after t_{16} (Check fee) occurs, according to the prescribed behavior in the process model, t_{17} (Review payment) and t_{18} (Delivery) should occur. However, in all three traces, t9 (Check goods) suddenly occurs, indicating that the buyer has unexpectedly increased or decreased the quantity of goods, prompting a recheck of the goods' quantity before deciding on sufficiency or insufficiency.



Check goods

	quantity(thousand)	supply ability	output
1	(0,3]	true	Sufficient goods
2	(3,5]	false	Insufficient goods
3	>5	false	Reject

transportation mode

	quantity(thousand)	transport ability	output
1	(0,2]	true	car
2	(2,5]	false	truck

Receive goods

	quality	rate（%）	output
1	True	100	Receive goods
2	False	(85,100)	Rework
3	False	(0,85]	Reject goods

Figure 4. Collaborative processes with decision analysis tables.

Assuming the initial intended purchase quantity by the buyer was 3000 units:

1. In the first trace, a decrease of 1000 units: After t_9 occurs, the quantity is still deemed sufficient. The order information is then sent to logistics. Due to the decrease in quantity, the transportation mode changes according to Figure 4, resulting in a change in freight. The seller then rechecks the fees. In this trace, the expected behaviors of t_9 are $\{t_{13}, t_{14}, t_{16}\}$, all of

which satisfy the rule for a quantity less than 3000 units. Therefore, the effective expected behaviors are the same as the expected behaviors.

2. In the second trace, an increase of 1000 units: After t_9 occurs, the quantity is insufficient at this time, and it needs to be purchased. The order information is then sent to logistics. With the increase in quantity, the transportation mode and freight also changes, so it is necessary to recheck the fees. In this trace, the expected behaviors of t_9 are {t_{11}, t_{12}, t_{14}, t_{16}}, all of which satisfy the rule for a quantity between 3000 and 5000 units. Thus, these are the effective expected behaviors.

3. In the third trace, an increase of 3000 units: The quantity is clearly insufficient, and procurement would be the next logical step in reality. The expected behaviors of t_9 in this scenario would typically include $\{t_{11}, t_{12}, t_{14}, t_{16}\}$. However, from the decision analysis table in Figure 4, it is found that the increased quantity does not meet the specified rule, and should be rejected, preventing any further actions. Consequently, in the third trace, t_9 has no effective expected behaviors.

3.3. Alignment Repair Based on Effective Expected Behavior

Process consistency analysis aims to identify and quantify the differences between the actual execution behavior (i.e., the event log) and the predefined process behavior (i.e., the process model) [38]. As two interrelated and independent entities, the event log and process model show a significant symmetry in consistency analysis. On the one hand, from the perspective of the event log, we carefully check every activity in the log to ensure that they can find a one-to-one corresponding execution path in the process model, which reflects the symmetry of mapping from log to model. On the other hand, based on the process model, we examine each activity in the model one by one to confirm whether they can find the actual records in the event log, which reflects the verification symmetry from the model to the log. This analysis is very important for ensuring the smooth implementation of business processes and finding potential problems in time. When the data changes, the collaborative process may be automatically adjusted to respond to these changes, in order to restore the execution trace to the normal process execution state. However, in practice, although some execution traces seem to deviate from the established process model, they are actually based on effective expected behavior in specific situations. These behaviors may align better with the needs of real-world operations; thus, simply treating them as anomalies or errors may not be appropriate. To more accurately reflect the execution of processes, it is necessary to identify these seemingly deviant but actually effective behaviors and repair the original alignment (i.e., the correspondence between execution traces and process models).

In Algorithm 2, Line 1 initializes the repaired alignment γ' to be a copy of the original alignment γ and the set of effective expected behaviors Θ' to be an empty set. Lines 2–20 iterate through each move in the optimal alignment. When a move is a model move, it checks if the corresponding activity a_i is in Θ'. If so, it modifies the corresponding >> in the repaired alignment to ρ, removes a_i from Θ', and, if not, the effective expected behavior of a_i is obtained by Algorithm 1 and added to Θ'. When a move is a log move, it checks if the corresponding event e_i is in Θ'. If so, it modifies the corresponding >> in the repaired alignment to ρ and removes e_i from Θ', and, if not, the effective expected behavior of e_i is obtained by using Algorithm 1 and added to Θ'. For a synchronous move, if a_i is in Θ', it removes a_i from Θ'. After iterating through all moves, the repaired alignment is achieved.

Algorithm 2. Repair Alignment

```
Input: original alignment $\gamma$, decision Petri net DN, trace $\sigma$
Output: alignment after repair $\gamma'$
1.  $\gamma' = \gamma, \Theta' = \varnothing$, i = 0
2.  for $i \le |\gamma|$ do
3.      for each move $(e_i, a_i) \in \gamma$ do
```

Algorithm 2. *Cont.*

```
        if $e_i$ =>> then
            if $a_i \in \Theta'$ then
                $\gamma'[i].e_i = \rho$
                $\Theta' \leftarrow \Theta' - \{a_i\}$
            else use Algorithm 1 to obtain the effective expected behavior of $a_i$ and add it to $\Theta'$
            end if
        else if $a_i$ =>> then
            if $e_i \in \Theta'$ then
                $\gamma'[i].a_i = \rho$
                $\Theta' \leftarrow \Theta' - \{e_i\}$
            else if use Algorithm 1 to obtain the effective expected behavior of $e_i$ and add it to $\Theta'$
            end if
        else if $e_i = a_i$ then
            if $a_i \in \Theta'$ then
                $\Theta' \leftarrow \Theta' - \{a_i\}$
            end if
        end if
    i++
    end for
    return $\gamma'$,$\Theta'$
```

Example 2. *Continue to analyze the three traces in the motivating example and repair the alignment in Table 1, and obtain Table 2 as follows:*

Table 1. Alignment.

σ_1	t_8	t_9	t_{13}	t_{14}	t_{15}	t_{16}	t_9	t_{13}	t_{14}	t_{16}	t_{17}	t_{18}	
Model	t_8	t_9	t_{13}	t_{14}	t_{15}	t_{16}	>>	>>	>>	>>	t_{17}	t_{18}	
σ_2	t_8	t_9	t_{13}	t_{14}	t_{15}	t_{16}	t_9	t_{11}	t_{12}	t_{14}	t_{16}	t_{17}	t_{18}
Model	t_8	t_9	t_{13}	t_{14}	t_{15}	t_{16}	>>	>>	>>	>>	>>	t_{17}	t_{18}
σ_3	t_8	t_9	t_{13}	t_{14}	t_{15}	t_{16}	t_9	t_{11}	t_{12}	t_{16}	t_{17}	t_{18}	
Model	t_8	t_9	t_{13}	t_{14}	t_{15}	t_{16}	>>	>>	>>	>>	t_{17}	t_{18}	

Table 2. Repair of alignment.

σ_1	t_8	t_9	t_{13}	t_{14}	t_{15}	t_{16}	t_9	t_{13}	t_{14}	t_{16}	t_{17}	t_{18}	
Model	t_8	t_9	t_{13}	t_{14}	t_{15}	t_{16}	>>	ρ	ρ	ρ	t_{17}	t_{18}	
σ_2	t_8	t_9	t_{13}	t_{14}	t_{15}	t_{16}	t_9	t_{11}	t_{12}	t_{14}	t_{16}	t_{17}	t_{18}
Model	t_8	t_9	t_{13}	t_{14}	t_{15}	t_{16}	>>	ρ	ρ	ρ	ρ	t_{17}	t_{18}
σ_3	t_8	t_9	t_{13}	t_{14}	t_{15}	t_{16}	t_9	t_{11}	t_{12}	t_{16}	t_{17}	t_{18}	
Model	t_8	t_9	t_{13}	t_{14}	t_{15}	t_{16}	>>	>>	>>	>>	t_{17}	t_{18}	

The fitness of the three traces and the model are calculated again to be 0.917, 0.923, and 0.67, respectively. It is found that σ_2, which had the lowest fitness, became the largest after repair, while σ_3 has no change in fitness, changing from the original maximum to the minimum. This indicates that in the fitness calculation based on optimal alignment, the original optimal alignment can be corrected for reasonable deviation, and then the fitness can be re-calculated. This is a response to unexpected situations and is also different from other consistency calculation methods.

4. Experimental Analysis and Evaluation

4.1. Experimental Setup

In this paper, two kinds of data sets are used for the experiment: an artificial data set and a real data set. The artificial data set is the artificial event log generated by PLG2.0, and the real data set is the event log in the public data set BPIC2015.

4.2. Experimental Process and Results

In order to evaluate the effectiveness, feasibility, and accuracy of the method proposed in this paper, the relevant experimental steps are as follows: firstly, the event log is mined by process mining technology to obtain the corresponding process model. Secondly, different degrees and types of noise are randomly inserted into the event log to generate event logs with different noise ratios and types. The inserted noise includes the following three situations: randomly inserting n events or deleting n events in the event sequence, randomly modifying the data value of the event, and a mixture of the two. The three noise types are evenly distributed according to the deviation ratio of 5%, 10%, and 15%. Then the deviation sequence is aligned with the process model, and the effective expected behavior of deviation activities and the consistency between the deviation sequence and the process model are obtained according to the algorithm. Finally, the effectiveness of our method is evaluated by comparing it with method1 [38] and method2 [39].

Analyze the manual event log: Compare the precision, recall, and F1-score of effective expected behavior with and without rule constraints when deviations occur.

$$Precision = \frac{TP}{TP + FP}$$

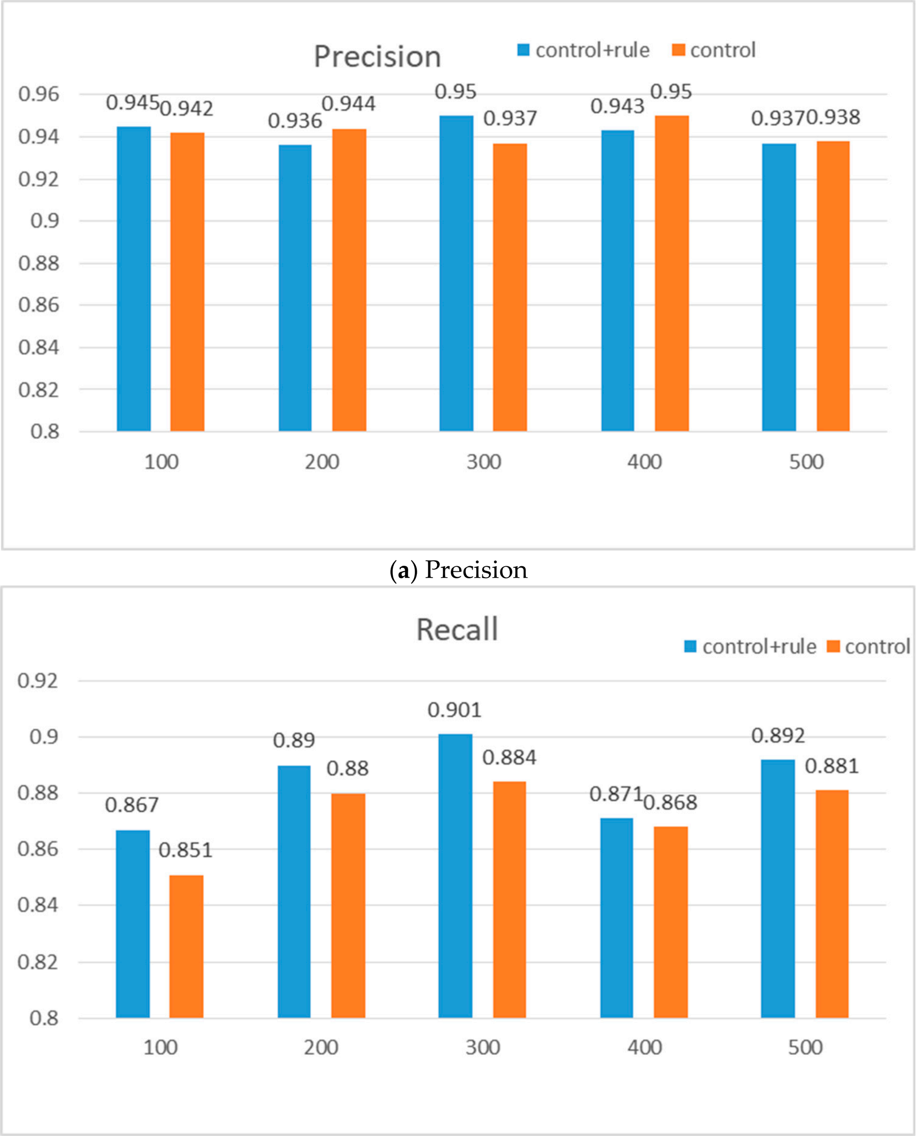
$$\text{Recall} = \frac{TP}{TP + FN}$$

$$F1 = \frac{2 * Precision * \text{Recall}}{Precision + \text{Recal}} \quad (1)$$

where TP is the number of positive samples judged as positive, and FN is the number of positive samples judged as negative. FP is the number of negative samples judged as positive.

From Figure 5, it can be seen that when the control flow changes, considering the rule constraints is generally better than not considering the rule constraints in terms of recall and F1-score, which indicates that considering the rule constraints is more comprehensive in identifying the effective expected behavior, and performs better in balancing accuracy and recall. Not considering rule constraints is close in accuracy to considering rule constraints, and even slightly higher in the first four points, but slightly lower in recall and F1-score than considering rule constraints. Therefore, the impact of rule constraints on the effectiveness of expected behavior is not great, but most of them are better than those without considering rule constraints.

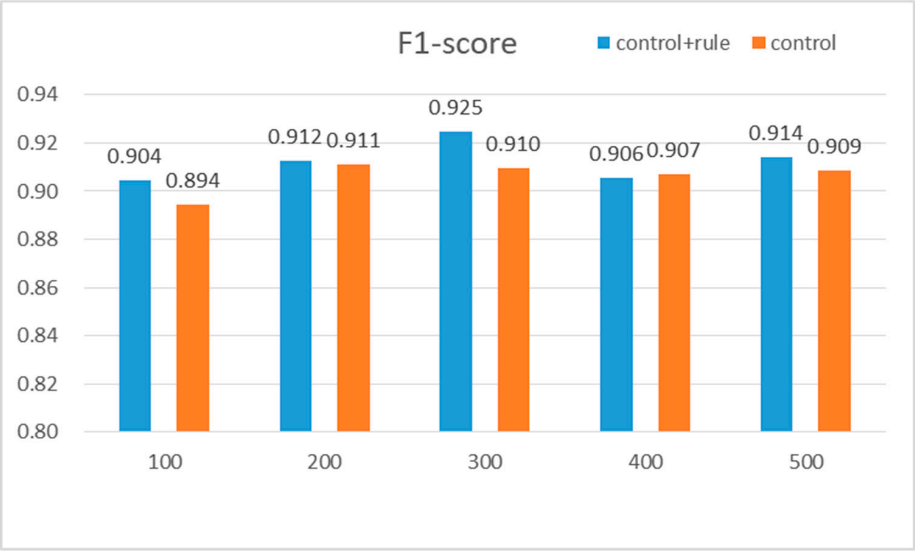
As can be seen from Figure 6, when the data value changes, in terms of accuracy, the average accuracy of the effective expected behavior considering the rule constraints is improved by about 4.7%, reaching a peak at 0.945, which means that the accuracy of the effectiveness of the obtained expected behavior is relatively high. The overall accuracy without considering the rule constraints is lower than that when considering the rule constraints. The lower accuracy may mean that there are more false positives when judging as positive, that is, the expected behavior that is not effective originally is judged as effective, which affects the reliability of the results. In terms of recall, the average recall of effective expected behavior increased by about 4% when considering the rule constraints, and the peak value reached 0.865. This indicates that the actual effective expected behavior can be found more effectively, and the situation of missing detection is reduced. The recall without considering the rule constraint is lower than that with considering the rule constraint. A lower recall may mean that not all positive samples are fully identified, and there are more false negatives. In terms of F1-score, the average F1-score of effective expected behavior is increased by 4.3% when considering the rule constraints, indicating the comprehensive performance of precision and recall is better. The increase in F1-score shows that the rule constraint not only improves the accuracy and recall, but also maintains the balance between them. The F1-score of the method without considering the rule constraint is generally lower than that of the method when considering the rule constraint. The lower F1-score indicates that the balance between precision and recall is poor, and that the comprehensive performance is weak without considering the rule constraint. Therefore, when analyzing the impact of data changes on the process, rule constraints should be considered to avoid identifying abnormal behavior as effective expected behaviors, thus reducing the false negative rate of effective expected behaviors and laying a foundation for the consistency analysis of event logs and models in the future.



(**a**) Precision

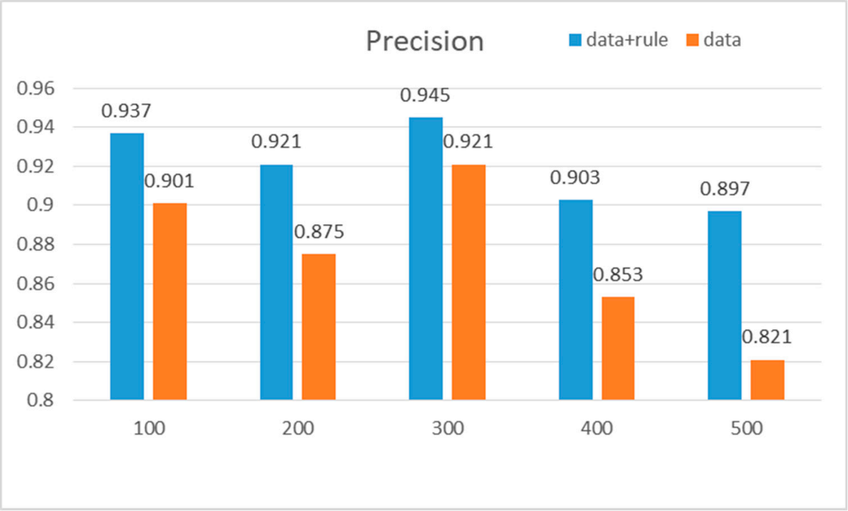
(**b**) Recall

Figure 5. *Cont.*

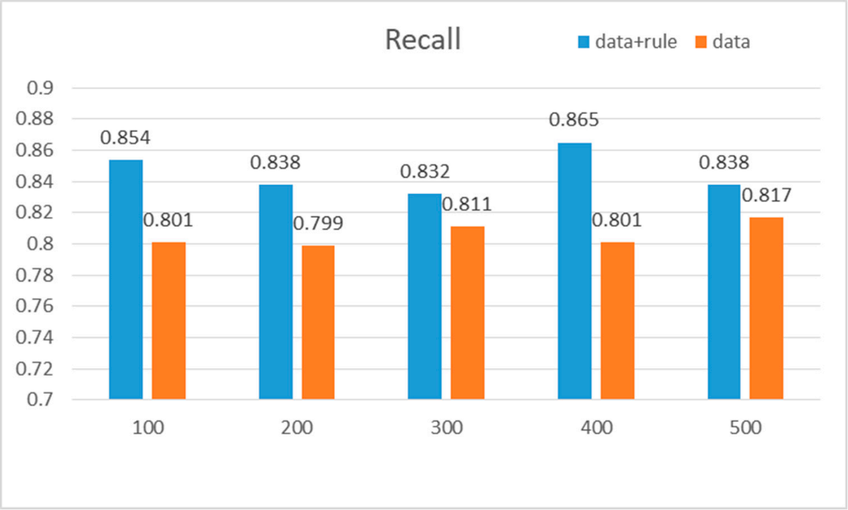


(**c**) F1-score

Figure 5. Control Flow and Rule Result Comparison.

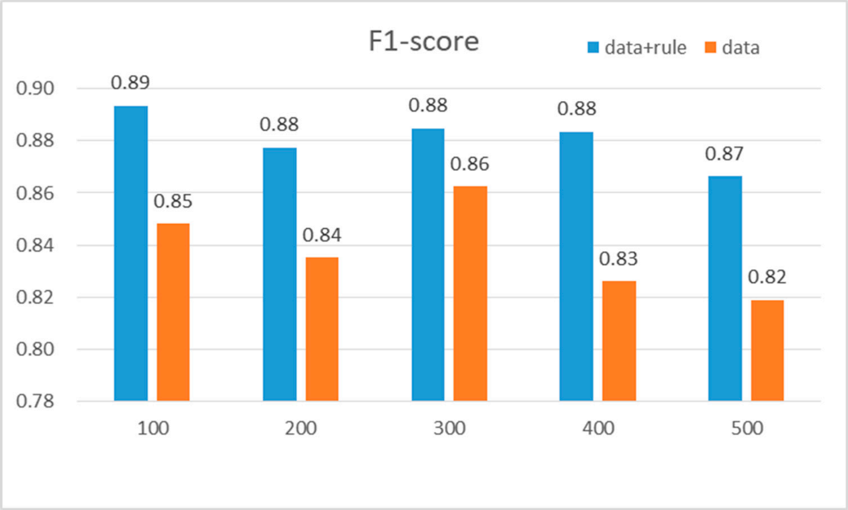


(**a**) Precision



(**b**) Recall

Figure 6. *Cont.*



(**c**) F1-score

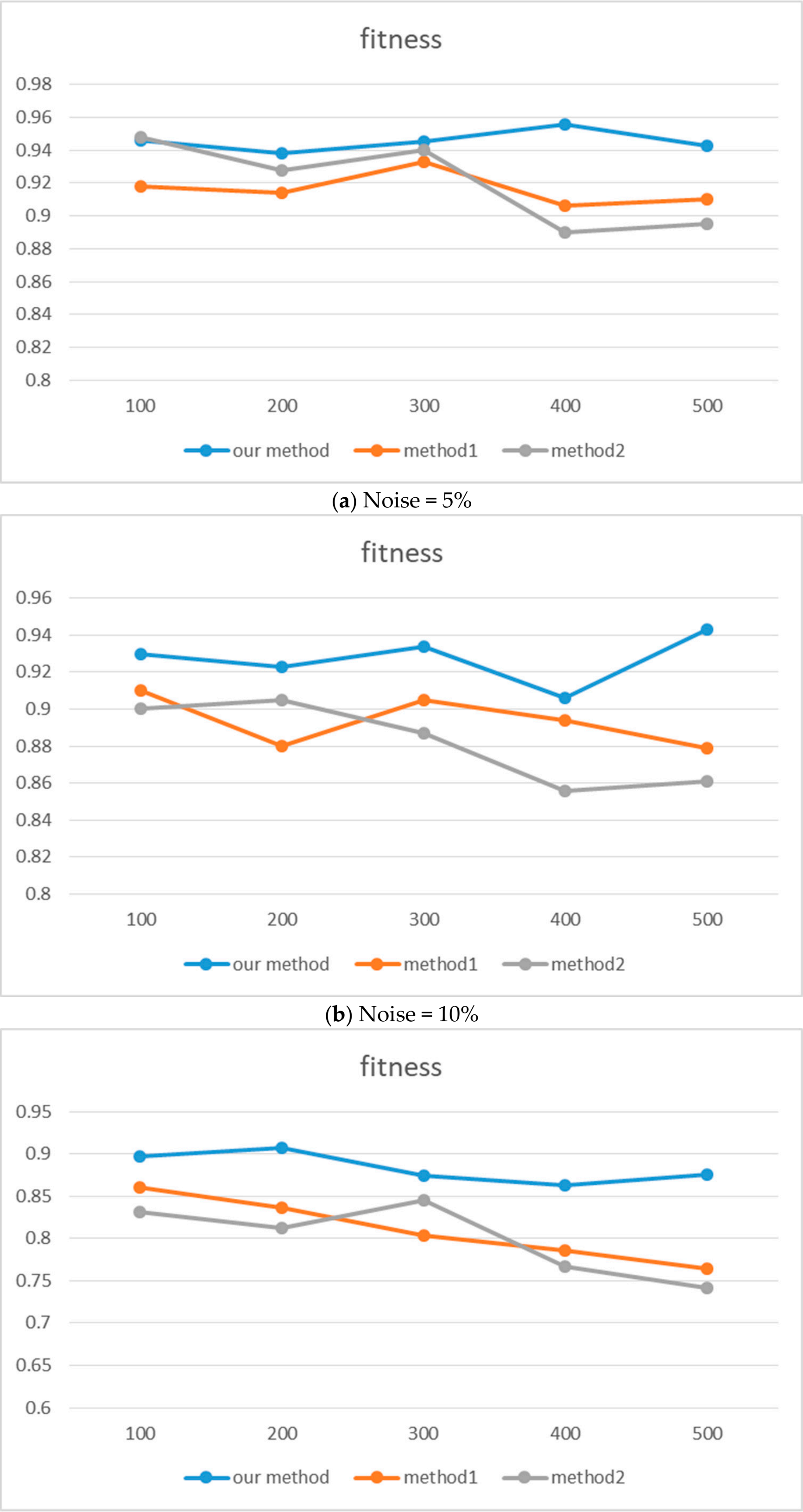
Figure 6. Data and rule result comparisons.

When comparing the consistency calculation method proposed in this paper (hereinafter referred to as "our method") with the existing method1 and method2, we use fitness as the evaluation index and make a systematic analysis based on the experimental results in Figures 7 and 8 and Table 3. Specifically, when faced with different levels of noise interference (5%, 10%, and 15%, respectively), the performance of the new method shows good robustness. In a 5% noise environment, compared with the method1, the average fitness of our method is improved by about 2.94% and 2.9% on the artificial and real event logs, respectively. Compared with method2, the growth rates are about 2.54% and 5.4%, respectively. This result preliminarily verifies the effectiveness of the new method in dealing with slight data disturbances. With the noise level increasing to 10%, compared with method1, the average fitness on artificial and real event logs is further improved by about 3.3% and 3.96%, respectively. Compared with method2, the improvement is about 3.96% and 6.78%, respectively. When the noise level reaches a high level of 15%, compared with method1, the average fitness of our method in artificial and real event logs is improved by about 7.3% and 8.3%, respectively. Even compared with method2, significant increases of about 4.5% and 7.1% were achieved, respectively. Based on the above data, it shows that our method has good robustness and adaptability under different noise conditions, and its overall adaptability is relatively stable. Although there is slight fluctuation, its overall performance is better than the other two methods, indicating that this method has good noise resistance and adaptability.

Table 3. Average fitness under different noise levels.

Fitness	**Noise = 5%**		**Noise = 10%**		**Noise = 15%**	
	Artificial Data	**Real Data**	**Artificial Data**	**Real Data**	**Artificial Data**	**Real Data**
Our method	94.56%	89.94%	92.72%	81.48%	88.34%	78.64%
Method1	91.62%	87.04%	89.36%	77.52%	81.04	74.54%
Method2	92.02%	84.54%	88.18%	74.7%	79.96%	71.5%

To sum up, the method proposed in this paper can effectively retrieve the expected behavior of deviation activities, reduce the false negative rate of effective expected behavior, and improve the consistency between the process and event log.

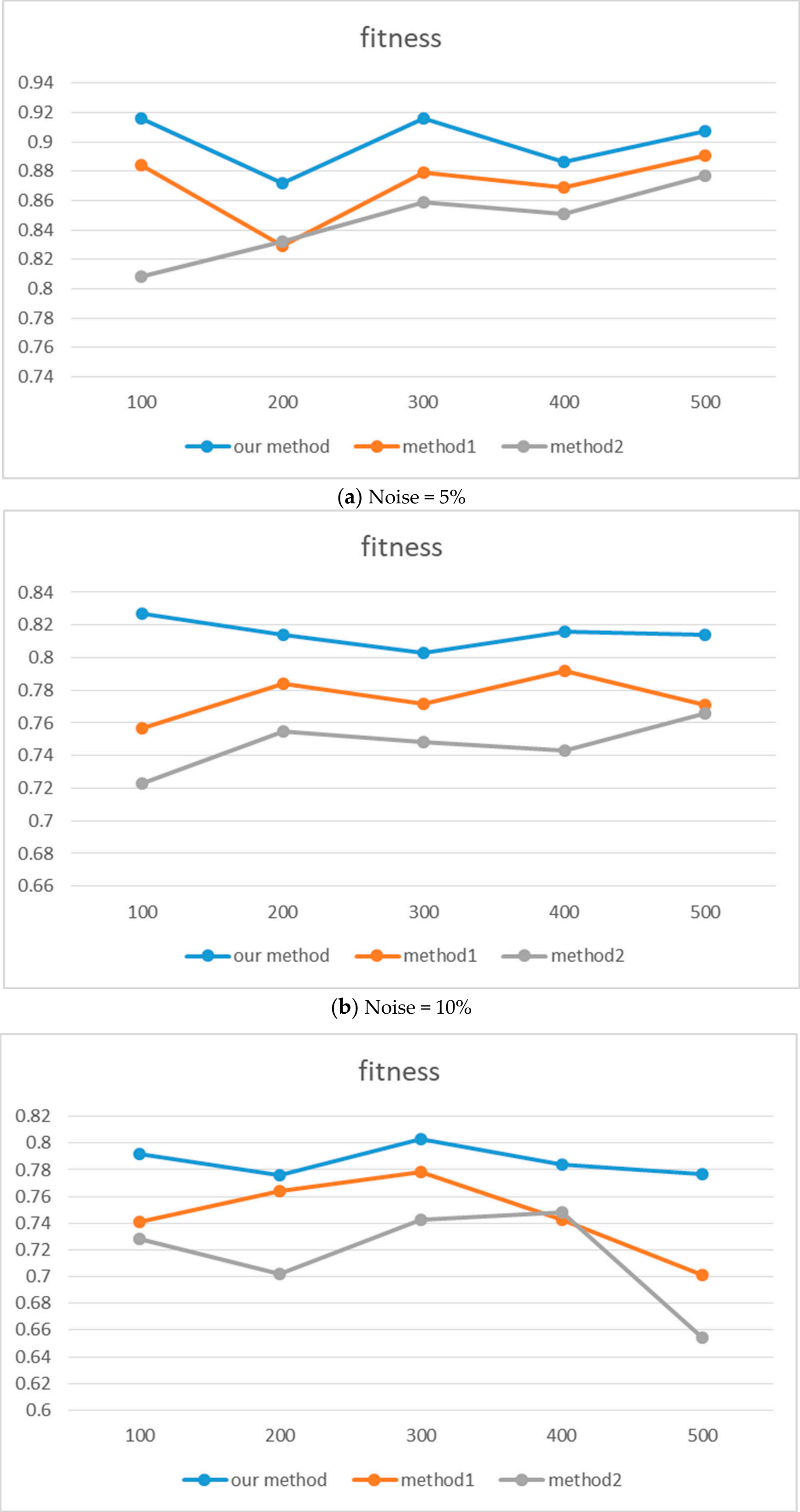


(**a**) Noise = 5%

(**b**) Noise = 10%

(**c**) Noise = 15%

Figure 7. Comparison of results with manual data sets.



(**a**) Noise = 5%

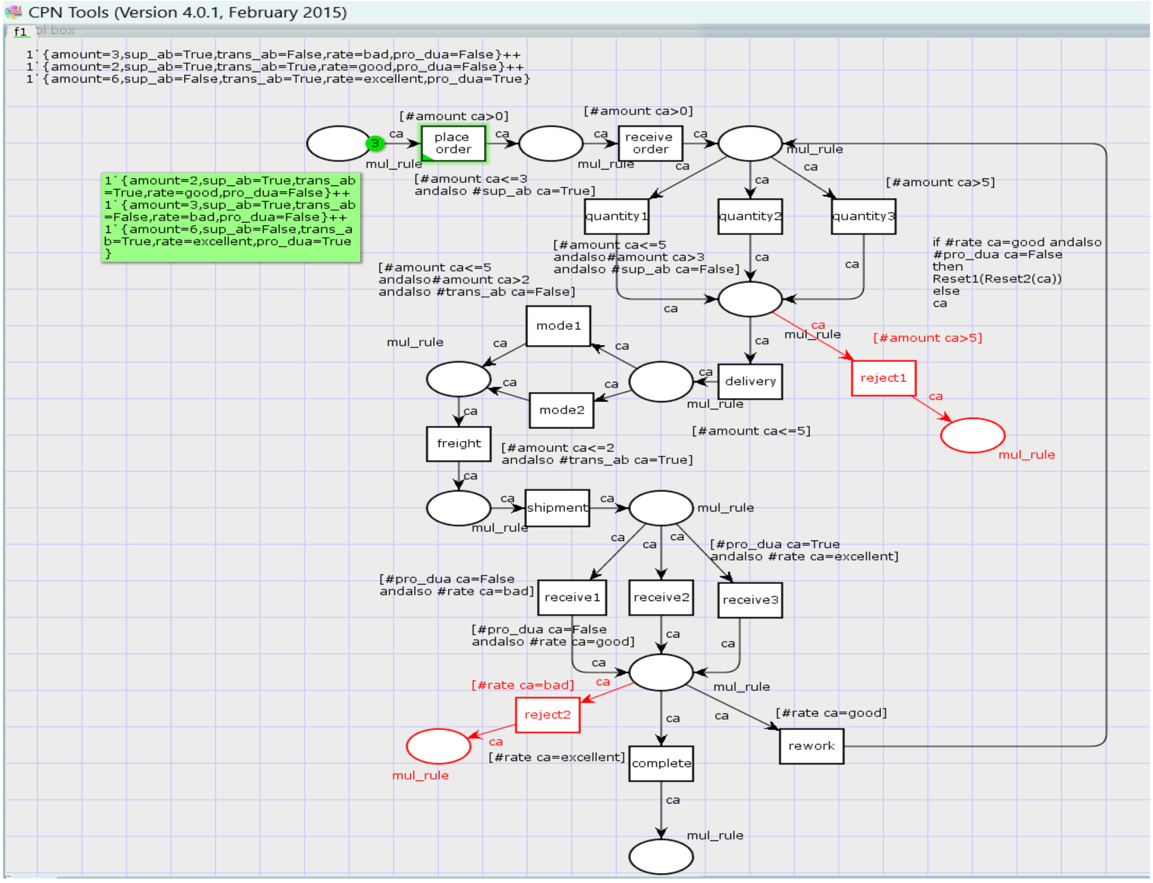
(**b**) Noise = 10%

(**c**) Noise = 15%

Figure 8. Comparison of results with real data sets.

4.3. CPN Tools Simulation

In addition, in order to show the impact of data changes, the change propagation process of the collaborative process is simulated based on CPN Tools. Figure 9 corresponds to the collaboration process with the decision analysis table in Figure 4, and the black part represents the basic collaboration process, including the control flow, data elements, and transition guards, where the decision analysis table in Figure 4 is transformed into transition guards in Figure 4. The green part represents the initial mark, and the model must contain the green part in order to run. For processes that contain data elements and complex rule constraints, there may be some defects, and the red part indicates the defect status under the initial marking.



CPN Tools (Version 4.0.1, February 2015)

```
1`{amount=3,sup_ab=True,trans_ab=False,rate=bad,pro_dua=False}++
1`{amount=2,sup_ab=True,trans_ab=True,rate=good,pro_dua=False}++
1`{amount=6,sup_ab=False,trans_ab=True,rate=excellent,pro_dua=True}
```

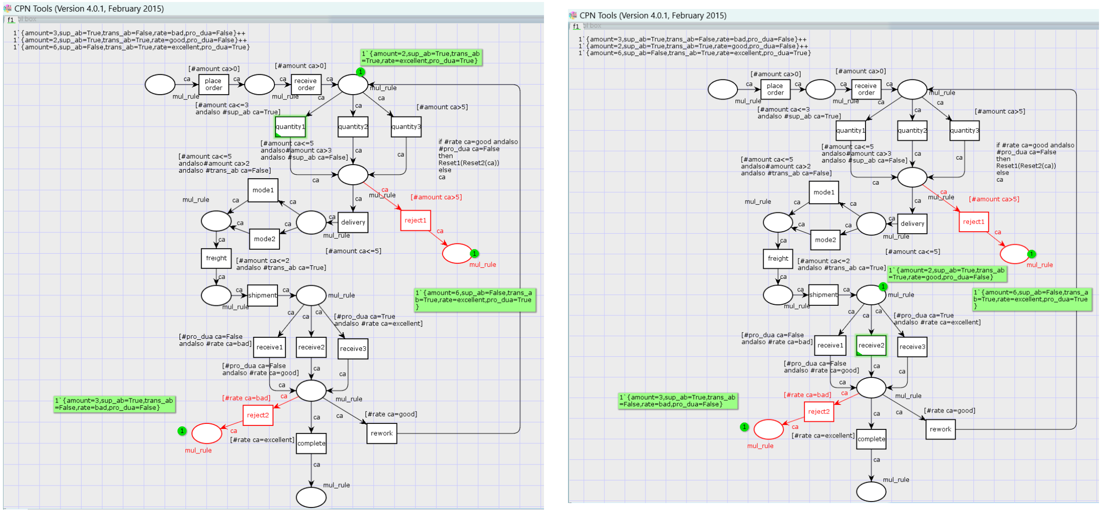
```
1`{amount=2,sup_ab=True,trans_ab
=True,rate=good,pro_dua=False}++
1`{amount=3,sup_ab=True,trans_ab
=False,rate=bad,pro_dua=False}++
1`{amount=6,sup_ab=False,trans_a
b=True,rate=excellent,pro_dua=True
}
```

```
[#amount ca>0]
[#amount ca>0]
[#amount ca<=3
andalso #sup_ab ca=True]
[#amount ca>5]
[#amount ca<=5
andalso#amount ca>3
andalso #sup_ab ca=False]
if #rate ca=good andalso
#pro_dua ca=False
then
Reset1(Reset2(ca))
else
ca
[#amount ca<=5
andalso#amount ca>2
andalso #trans_ab ca=False]
[#amount ca>5]
[#amount ca<=5]
[#amount ca<=2
andalso #trans_ab ca=True]
[#pro_dua ca=True
andalso #rate ca=excellent]
[#pro_dua ca=False
andalso #rate ca=bad]
[#pro_dua ca=False
andalso #rate ca=good]
[#rate ca=bad]
[#rate ca=good]
[#rate ca=excellent]
```

Figure 9. Collaboration process in CPN Tools.

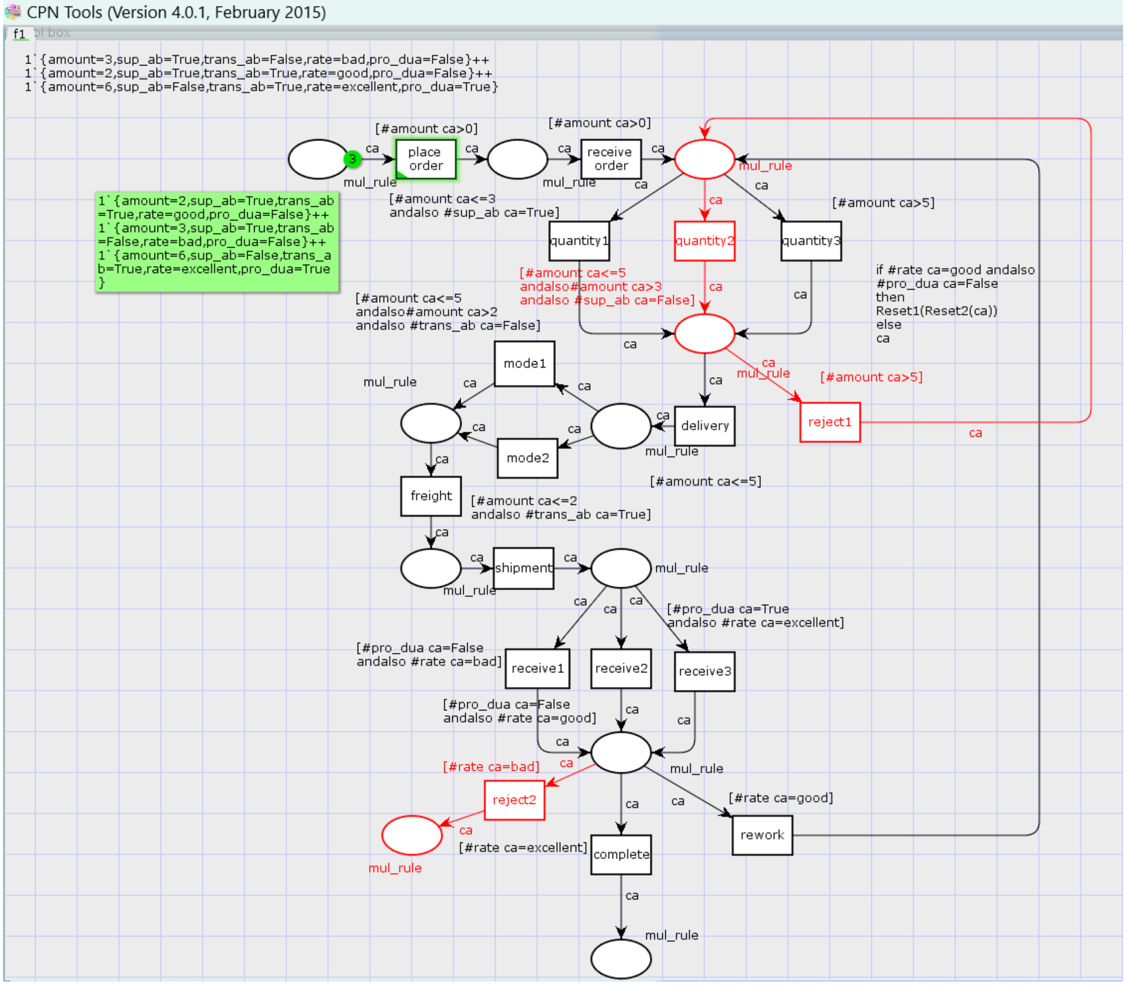
Figure 10a,b simulate the path trends of collaborative processes under different data elements. Figure 10a corresponds to that when the quality of goods is “False” and the qualified rate is “good”, according to the description of the decision analysis table, “receive2” should be selected at this time, and if the goods quality becomes “True” and the qualified rate is “excellent” after rework, “receive3” should be selected. Here, in order to better express the qualified rate, 100% is equivalent to “excellent”, 85–100% is equivalent to “good”, and less than 85% is equivalent to “bad”. Figure 10b shows that when the quantity of goods is changed from 3000 to 2000, and the corresponding constraint rules are also changed, it is also reasonable to execute quantity1-Mode2-Receive2, and the deviation activity in the trace is $\{t_9\}$.

Figure 11 describes the impact of data changes on process execution behavior and continues to analyze the motivating example. When the number of goods increases by 3000, the process execution should choose “quantity3”—reject1, but actually executes quantity2-delivery in the trace. Obviously, if the data constraint rules are not met, the change propagation cannot be carried out smoothly, so t_9 in the trace has no effective expected behavior, and the deviation activities are $\{t_9, t_{13}, t_{14}, t_{16}\}$.


CPN Tools (Version 4.0.1, February 2015)
CPN Tools (Version 4.0.1, February 2015)
place order
receive order
quantity1
quantity2
quantity3
mode1
mode2
delivery
reject1
freight
shipment
receive1
receive2
receive3
reject2
complete
rework
mul_rule
ca
1`{amount=2,sup_ab=True,trans_ab=True,rate=excellent,pro_dua=True}
1`{amount=6,sup_ab=False,trans_ab=True,rate=excellent,pro_dua=True}
1`{amount=3,sup_ab=True,trans_ab=False,rate=bad,pro_dua=False}
1`{amount=2,sup_ab=True,trans_ab=True,rate=good,pro_dua=False}

(**a**) (**b**)

Figure 10. Initial marking changes in the collaboration process.


CPN Tools (Version 4.0.1, February 2015)

```
1`{amount=3,sup_ab=True,trans_ab=False,rate=bad,pro_dua=False}++
1`{amount=2,sup_ab=True,trans_ab=True,rate=good,pro_dua=False}++
1`{amount=6,sup_ab=False,trans_ab=True,rate=excellent,pro_dua=True}
```

```
1`{amount=2,sup_ab=True,trans_ab
=True,rate=good,pro_dua=False}++
1`{amount=3,sup_ab=True,trans_ab
=False,rate=bad,pro_dua=False}++
1`{amount=6,sup_ab=False,trans_a
b=True,rate=excellent,pro_dua=True
}
```

Figure 11. Data changes in the collaboration process.

5. Conclusions and Future

In the field of business process management, process change analysis is one of the core topics. Researchers have put forward various methods to analyze the impact of change, but the existing research mainly focuses on the change analysis at the control flow level, ignoring the impact of data changes. In view of this, this paper innovatively puts forward a method to evaluate and repair the consistency of collaborative process data changes with rule constraints. Firstly, this method deeply analyzes how data changes affect other components in the process and determines the expected behavior of deviation activities affected by data changes under the optimal process alignment state through a refined analysis. Subsequently, the decision rules are introduced as the evaluation framework to evaluate the rationality and effectiveness of these expected behaviors in order to screen out the set of effective expected behaviors that conform to the logic and business rules. Based on this set, the repair strategy of process alignment is further designed, aimed at adjusting and optimizing the initial optimal alignment state, ensuring that the repaired process not only solves the problems caused by data changes, but also improves the consistency of the process. The experimental simulation data verify the advantages of this method in accurately identifying expected behaviors and improving process consistency. The average accuracy, recall, and F1-score of effective expected behaviors with rule constraints are improved by 4%, 4.7%, and 4.3%, respectively, and the average fitness values of artificial event logs/real event logs with different noise levels are 94.56%/89.94%, 92.72%/81.48%, and 88.34%/78.64%, respectively. It is worth noting that the research in this paper starts with optimal alignment, which has certain limitations. In the future, we can study the consistency of process behavior related to data and guard change, and then analyze the impact of data change and guard change on process consistency and apply it to electronic payment systems and data leakage to improve security.

Author Contributions: Conceptualization, Q.W.; methodology, Q.W.; software, Q.W. and C.S.; validation, C.S.; formal analysis, C.S.; investigation, Q.W.; resources, Q.W.; data curation, Q.W. and C.S.; writing—original draft preparation, Q.W.; writing—review and editing, Q.W. and C.S.; visualization, Q.W. and C.S.; supervision, Q.W. and C.S.; project administration, Q.W.; funding acquisition, Q.W. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the National Natural Science Foundation of China, grant number 61572035, 61402011; the Key projects of natural science research in Anhui Universities, grant number 2022AH051638, 2022AH051651, 2023AH051865; the Anhui Agriculture Research System, the Project of Anhui Provincial Department of Science and Technology, grant number 202204c06020065; the Key Discipline Construction Project of Anhui Science and Technology University, grant number XK-XJGY002; and The APC was funded by the Key Discipline Construction Project of Anhui Science and Technology University, grant number XK-XJGY002.

Data Availability Statement: The data sets used in the study are from the 4TU.Centre for Research Data, available online at https://data.4tu.nl/datasets/372d0cad-3fb1-4627-8ea9-51a09923d331/1 (accessed on 8 March 2024).

Acknowledgments: The authors would like to express their gratitude to the editor and anonymous reviewers for their valuable comments and constructive suggestions on the original manuscript.

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Dai, W.; Covvey, D.; Alencar, P.; Cowan, D. Lightweight Query-Based Analysis of Workflow Process Dependencies. *J. Syst. Softw.* **2009**, *82*, 915–931. [CrossRef]
2. Sidorova, N.; Stahl, C.; Trčka, N. Soundness Verification for Conceptual Workflow Nets with Data: Early Detection of Errors with the Most Precision Possible. *Inf. Syst.* **2011**, *36*, 1026–1043. [CrossRef]
3. Liu, B.; Hsu, W.; Han, H.-S.; Xia, Y. Mining Changes for Real-Life Applications. In Proceedings of the Data Warehousing and Knowledge Discovery; Kambayashi, Y., Mohania, M., Tjoa, A.M., Eds.; Springer: Berlin/Heidelberg, Germany, 2000; pp. 337–346.
4. van der Aalst, W.M.P.; Weijters, A.J.M.M. Process Mining: A Research Agenda. *Comput. Ind.* **2004**, *53*, 231–244. [CrossRef]

5. Günther, C.W.; Rinderle, S.; Reichert, M.; Van Der Aalst, W. Change Mining in Adaptive Process Management Systems. In *On the Move to Meaningful Internet Systems 2006: CoopIS, DOA, GADA, and ODBASE*; Meersman, R., Tari, Z., Eds.; Lecture Notes in Computer Science; Springer Berlin Heidelberg: Berlin/Heidelberg, Germany, 2006; Volume 4275, pp. 309–326, ISBN 978-3-540-48287-1.
6. Fang, H.; Sun, S.Y.; Fang, X.W. Behavior change mining methods based on incomplete logs conjoint occurrence relation. *CIMS* **2020**, *26*, 1887–1895.
7. Sun, S.; Li, Q. A Behavior Change Mining Method Based on Complete Logs with Hidden Transitions and Their Applications in Disaster Chain Risk Analysis. *Sustainability* **2023**, *15*, 1655. [CrossRef]
8. Fang, H.; Zhang, Y.; Wu, Q.L. A log induced change mining method for fault diagnosis using structure causality in BPMSs. *Control. Theor. Technol.* **2018**, *35*, 1167–1176.
9. Hmami, A.; Sbai, H.; Fredj, M. A New Framework to Improve Change Mining in Configurable Process. In Proceedings of the Proceedings of the 3rd International Conference on Networking, Information Systems & Security, Marrakech, Morocco, 31 March–2 April 2020; Association for Computing Machinery: New York, NY, USA, 2020; pp. 1–6.
10. Hmami, A.; Sbai, H.; Fredj, M. Enhancing Change Mining from a Collection of Event Logs: Merging and Filtering Approaches. *J. Phys. Conf. Ser.* **2021**, *1743*, 012020. [CrossRef]
11. Weidlich, M.; Mendling, J.; Weske, M. Propagating Changes between Aligned Process Models. *J. Syst. Softw.* **2012**, *85*, 1885–1898. [CrossRef]
12. Zhang, F.; Fang, X.W.; Fang, H. Analysis Method of the Smallest Change Region with Dynamic Slice of Petri Nets. *JFCS* **2016**, *10*, 516–523.
13. Fang, X.W.; Zhao, F.; Liu, X.W.; Fang, H. Change Propagation Analysis in Business Process Based on Behavior Inclusion and Behavior Inheritance of Petri. *SJCS* **2016**, *43*, 36–39.
14. Fang, X.; Liu, L.; Liu, X. Analyzing Method of Change Region in BPM Based on Module of Petri Net. *Inf. Technol. J.* **2013**, *12*, 1655–1659. [CrossRef]
15. Zhao, F.; Xiang, D.; Liu, G.; Jiang, C. Behavioral Consistency Measurement between Extended WFD-Nets. *Inf. Syst.* **2023**, *119*, 102274. [CrossRef]
16. Mahfouz, A.; Barroca, L.; Laney, R.; Nuseibeh, B. Requirements-Driven Collaborative Choreography Customization. In Proceedings of the Service-Oriented Computing; Baresi, L., Chi, C.-H., Suzuki, J., Eds.; Springer: Berlin/Heidelberg, Germany, 2009; pp. 144–158.
17. Weidlich, M.; Weske, M.; Mendling, J. Change Propagation in Process Models Using Behavioural Profiles. In Proceedings of the 2009 IEEE International Conference on Services Computing, Bangalore, India, 21–25 September 2009; pp. 33–40.
18. Kolb, J.; Kammerer, K.; Reichert, M. Updatable Process Views for User-Centered Adaption of Large Process Models. In Proceedings of the Service-Oriented Computing; Liu, C., Ludwig, H., Toumani, F., Yu, Q., Eds.; Springer: Berlin/Heidelberg, Germany, 2012; pp. 484–498.
19. Fdhila, W.; Rinderle-Ma, S.; Baouab, A.; Perrin, O.; Godart, C. On Evolving Partitioned Web Service Orchestrations. In Proceedings of the 2012 Fifth IEEE International Conference on Service-Oriented Computing and Applications (SOCA), Taipei, Taiwan, 17–19 December 2012; pp. 1–6.
20. Dahman, K.; Charoy, F.; Godart, C. Alignment and Change Propagation between Business Processes and Service-Oriented Architectures. In Proceedings of the 2013 IEEE International Conference on Services Computing, Santa Clara, CA, USA, 28 June–3 July 2013; pp. 168–175.
21. Wootton, J.C. Introduction to Computational Biology: Maps, Sequences and Genomes; Interdisciplinary Statistics. *Comput. Chem.* **1997**, *21*, 275–278. [CrossRef]
22. Wang, Y.; Yang, J.; Zhao, W.; Su, J. Change Impact Analysis in Service-Based Business Processes. *SOCA* **2012**, *6*, 131–149. [CrossRef]
23. Dam, H.K.; Ghose, A. Mining Version Histories for Change Impact Analysis in Business Process Model Repositories. *Comput. Ind.* **2015**, *67*, 72–85. [CrossRef]
24. Suvorov, N.M.; Lomazova, I.A. Verification of Data-Aware Process Models: Checking Soundness of Data Petri Nets. *J. Log. Algebr. Methods Program.* **2024**, *138*, 100953. [CrossRef]
25. Xiang, D.; Liu, G.; Yan, C.; Jiang, C. Detecting Data-Flow Errors Based on Petri Nets with Data Operations. *IEEE/CAA J. Autom. Sin.* **2018**, *5*, 251–260. [CrossRef]
26. Zhang, X.; Song, W.; Wang, J.; Xing, J.; Zhou, Q. Measuring Business Process Consistency Across Different Abstraction Levels. *IEEE Trans. Netw. Serv. Manag.* **2019**, *16*, 294–307. [CrossRef]
27. Tsoury, A.; Soffer, P.; Reinhartz-Berger, I. Data Impact Analysis in Business Processes. *Bus. Inf. Syst. Eng.* **2020**, *62*, 41–60. [CrossRef]
28. Tsoury, A.; Soffer, P.; Reinhartz-Berger, I. Impact-Aware Conformance Checking. In Proceedings of the Business Process Management Workshops; Di Francescomarino, C., Dijkman, R., Zdun, U., Eds.; Springer International Publishing: Cham, Switzerland, 2019; pp. 147–159.
29. van der Aalst, W.; Weijters, T.; Maruster, L. Workflow Mining: Discovering Process Models from Event Logs. *IEEE Trans. Knowl. Data Eng.* **2004**, *16*, 1128–1142. [CrossRef]
30. Weijters, A.J.M.M.; Ribeiro, J.T.S. Flexible Heuristics Miner (FHM). In Proceedings of the 2011 IEEE Symposium on Computational Intelligence and Data Mining (CIDM), Paris, France, 11–15 April 2011; pp. 310–317.

31. van Eck, M.L.; Buijs, J.C.A.M.; van Dongen, B.F. Genetic Process Mining: Alignment-Based Process Model Mutation. In Proceedings of the Business Process Management Workshops; Fournier, F., Mendling, J., Eds.; Springer International Publishing: Cham, Switzerland, 2015; pp. 291–303.
32. Bogarin, A.; Cerezo, R.; Romero, C. Discovering Learning Processes Using Inductive Miner: A Case Study with Learning Management Systems (LMSs). *Psicothema* **2018**, *30*, 322–330. [PubMed]
33. Mannhardt, F.; de Leoni, M.; Reijers, H.A.; van der Aalst, W.M.P. Balanced Multi-Perspective Checking of Process Conformance. *Computing* **2016**, *98*, 407–437. [CrossRef]
34. Wang, L.-L.; Fang, X.-W.; Shao, C.-F.; Asare, E. An Approach for Mining Multiple Types of Silent Transitions in Business Process. *IEEE Access* **2021**, *9*, 160317–160331. [CrossRef]
35. Bazhenova, E.; Buelow, S.; Weske, M. Discovering Decision Models from Event Logs. In Proceedings of the Business Information Systems; Abramowicz, W., Alt, R., Franczyk, B., Eds.; Springer International Publishing: Cham, Switzerland, 2016; pp. 237–251.
36. der Aalst, W.V. Process Mining: Data Science in Action. In *Process Mining: Data Science in Action*; Springer: Berlin/Heidelberg, Germany, 2016; pp. 1–467.
37. Song, W.; Jacobsen, H.-A. Static and Dynamic Process Change. *IEEE Trans. Serv. Comput.* **2018**, *11*, 215–231. [CrossRef]
38. Adriansyah, A.; van Dongen, B.F.; van der Aalst, W.M. Memory-Efficient Alignment of Observed and Modeled Behavior. *BPM Cent. Rep.* **2013**, *3*, 1–44.
39. Bergami, G.; Maggi, F.M.; Marrella, A.; Montali, M. Aligning Data-Aware Declarative Process Models and Event Logs. In Proceedings of the Business Process Management; Polyvyanyy, A., Wynn, M.T., Van Looy, A., Reichert, M., Eds.; Springer International Publishing: Cham, Switzerland, 2021; pp. 235–251.

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.