sustainability

MDPI

Article

A Behavior Change Mining Method Based on Complete Logs with Hidden Transitions and Their Applications in Disaster Chain Risk Analysis

Shuya Sun [1,2] and Qingsheng Li [3,*]

1 Research Institute for Risk Governance and Emergency Decision Making, School of Management Science and Engineering, Nanjing University of Information Science & Technology, Nanjing 210044, China
2 School of Mathematics and Finance, Chuzhou University, Chuzhou 239000, China
3 School of Business, Linyi University, Linyi 276005, China
* Correspondence: liqingsheng@lyu.edu.cn

Abstract: The aim of change mining is to discover changes in process models based on execution data recorded in event logs. There may be hidden transitions in the process models related to, for example, business integration and user requirements that do not exist in event logs. Behavioral change mining in the case of hidden transitions is a fundamental problem in the field of change mining. Existing research on change mining has not considered the effects of hidden transitions. This paper proposes a novel method based on complete logs with hidden transitions for mining behavioral changes. We analyze the behavioral relations of activities based on changed logs under the condition that the original model is unknown. Log-driven change mining is realized by calculating the log behavioral profile, minimum successor relation, and log-weighted coefficient, which allows the mining of hidden transitions, as well as changed behavioral relations. Finally, this method is applied to disaster chain risk analysis, and the evolution of disaster chains in different scenarios is mined from disaster logs to determine the type of disaster chain. The results of this paper provide a scientific basis for the strategy of chain-cutting disaster mitigation in the emergency management of disaster chains.

Keywords: change mining; complete logs; behavioral profile; log minimum successor relation; disaster evolution analysis

check for updates

Citation: 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. https://doi.org/10.3390/su15021655

Academic Editors: Jie Li, Xiaobing Yu and Mei Cai

Received: 11 November 2022
Revised: 4 January 2023
Accepted: 11 January 2023
Published: 14 January 2023

Copyright: © 2023 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (https://creativecommons.org/licenses/by/4.0/).

1. Introduction

With the continuous innovation and development of theory and workflow technology in business process management, an increasing number of large enterprises and management organizations are using process models to formalize internal business processes. However, business process models are constantly changing with the successful development of enterprise management and changes in regulations, policies, user requirements, and software maintenance [1]. These changing factors affect the behavior and structure of the models. Therefore, correctly identifying and mining process model changes is important for fault detection and diagnosis and the maintenance of business systems [2].

1.1. Problem Statement and Research Questions

Previous studies on changes in business process models have mainly focused on the change region, change propagation, change logs, and anomaly detection. In recent years, many scholars have focused on detecting the differences between different models in the change region and the change propagation analysis of process models [3–5]. Meanwhile, in the case that the process model is out of sync and inconsistent, the change queue has been used to propagate changes and achieve synchronous continuity. The model change analysis technology proposed in this paper is beneficial for the detection of the abnormal behavior of process models in actual operation. Previously, a submodel of the labelled business process model was constructed on the basis of analyzing the behavioral characteristics of Petri nets, and the concept of the action mode was proposed to determine the minimum change region [3].

Once the model changes, model management technology becomes indispensable. A synchronization method based on model elements [4] has been used for change management between process models at different levels of abstraction. A method of change propagation [5] that detects existing changes between different models has been proposed to determine the change region and solve the inconsistency of a model in layered refinement. Fang et al. proposed a dynamic analysis method based on Petri net modules to search for the change region [6] and introduced the T invariant to optimize the method and thus obtain the change region most accurately. Their method also improved the accuracy and excellence of the model change detection. A change propagation technology based on software maintenance [7] has been proposed to provide a powerful tool in solving the change propagation of process models in software maintenance and evolution. In addition to the research on the change region and change propagation, several scholars have analyzed change fragments in process models in recent years [8–10]. Cross-organization collaboration becomes more complex as business requirements become more complex. The concept of configurable process fragments has been proposed to solve the changes in the complexity of similar process models [8]. Here, event logs are used as input to construct a configurable process model that contains change fragments. A method is presented which is based on the improved process structure tree and software product line [9] to automatically identify reusable variable fragments and merge similar fragments into main fragments. This method mainly starts from the change fragments and restructures the design of the model. Process fragments have been defined [10], and an improved fragment called the morphological fragment has been proposed for detecting process model changes and reusing process combinations. In the case that the original reference model is unknown [11], the morphological fragment provides a new cornerstone for research on business process change mining whereby business process changes can be discovered from the joint occurrence relations of incomplete event logs. BINet, a neural network architecture for real-time multi-perspective anomaly detection in business process event logs, has been introduced [12]. Additionally, a set of heuristics has been proposed for automatically setting the threshold of an anomaly-detection algorithm automatically. A novel approach to detecting event log anomalies in process event streams has been described [12]. This approach focuses on a general framework into which different anomaly-detection methods can be plugged. As early as 2018, Fang et al. investigated log mining model changes [13], but they compared system logs with the actual models to obtain possible change patterns.

In recent years, an increasing number of scholars have researched business process anomaly detection [12,14], workaround detection methods [15], and business process monitoring techniques [16]. Nolle et al. proposed a method [14] that uses autoencoders to detect and analyze anomalies. This method has been further refined in terms of performance and evaluated on more sophisticated datasets. Anomaly detection also focuses on detecting temporal workarounds in business processes. Weinzierl et al. proposed a deep-learning-based method [15] that helps to detect different workaround types in event logs. The method is accurate and functions most effectively in business processes, having fewer variations and a higher number of different activities. A technique for explainable predictive process monitoring was proposed in [16], where relevance values for process activities are extracted from a graph neural network. This method is the first to use gated graph neural networks (GGNNs) to make decisions more explainable. In short, anomaly detection research mainly focuses on business process detection and workaround detection. Change mining is a branch in data mining [17] that focus on researching the behavioral changes caused by changed logs. These are widespread research branches in business process management. By identifying outliers, researchers can gain important knowledge and make better data decisions.

A Petri net is also widely used in practical application scenarios of risk assessment and disaster chain analysis since this method dynamically simulates the evolution of disaster chains as a discrete modeling tool. Examples of its application can be seen for earthquakes [18], coal mining [19], floods [20], and oil spills [21]. A colored Petri net demonstrates the recovery process by simulating the behavior of a disaster and capturing the differences in resource allocation following an earthquake disaster [18]. At the same time, Petri net technology addresses delays in decision making, responses to conflicts, and limited resources during emergencies [19]. The construction of an evolutionary analysis model of a flood disaster chain [20] not only allows an analysis of disaster elements but also helps to quickly and effectively curb the spread of a disaster. These benefits are similar to those of an oil spill disaster chain model [21], with both models providing a scientific basis for a chain break mitigation strategy of disaster chain emergency management through a dynamic derivation of the evolutionary process of the disaster and an analysis of the maximum risk path.

However, previous studies on disaster chain risk analysis require a known Petri net model. In contrast, the actual reference model is unknown or contains hidden transitions in most business processes. The hidden transition is also called the invisible transition, and it mainly plays the role of routing. Research on the mining of business process changes is based on models [3–10] or logs [11–13,22] and fails to consider the effects of hidden transitions on model changes. In this paper, we propose a behavioral-change-mining method considering hidden transitions to improve existing methods under the condition of complete logs. We demonstrate the superior performance of our proposed method in experiments.

1.2. Contributions and Research Method

The innovations of the present study are summarized as follows.

- The concepts of the log minimum successor relation and log-weighted coefficient are proposed to quantify the occurrence dependencies of pairs of activities in logs.
- The behavioral relations of pairs of activities are quantitatively analyzed by calculating the log behavioral profiles and the log minimum successor relations. Additionally, the behavioral relations of pairs of activities are further refined.
- A change-mining method based on hidden transitions is proposed to mine different kinds of hidden transitions and solve the problem of searching for behavioral relation changes with hidden transitions in event logs.

The remainder of the paper is arranged as follows. Section 2 introduces a motivating example. Section 3 presents related knowledge on the event log, log behavioral profile, and log minimum successor relation. Section 4 constructs change-mining methods based on complete logs and presents examples. Section 5 presents an experiment to verify the correctness of the method. Section 6 describes a specific disaster case to validate the practicability of the proposed method. Section 7 concludes the paper with suggestions for future work.

2. Motivating Example

A hidden transition is also called an invisible transition. Since hidden transitions do not appear in any event trace, it is a challenging problem to discover behavioral changes that contain hidden transitions in the field of change mining. A complete log is a log that contains all transition activities and possible behavioral relations among all activities. In what follows, we illustrate the necessity of the change-mining method proposed in this paper by giving a detailed example.

Firstly, it is necessary to understand the history logs and actual logs of the system. To simplify notation, the history logs are denoted L_{his}, the actual logs are denoted L_{act}, a mapped letter set for logs is denoted Y_L, and the log behavioral profiles are denoted BP_L. For complete event logs $L_{his} = \{< ABD >, < ACD >\}$ and $L_{act} = \{< ABCD >, < ACBD >\}$, it is clear that $Y_{L_{his}}$ is equal to $Y_{L_{act}}$. If the mapping letters of the history logs and actual logs are equal, we need to further calculate the log behavioral profiles of $BP_{L_{his}}$ and $BP_{L_{act}}$. Behavioral profile relations are shown in Tables 1 and 2.

Table 1. Behavioral profile relations ($BP_{L_{his}}$).

	A	B	C	D
A		$\rightarrow$	$\rightarrow$	$\rightarrow$
B			$\rightarrow$	$\rightarrow$
C				$\rightarrow$
D				

Table 2. Behavioral profile relations ($BP_{L_{act}}$).

	A	B	C	D
A		$\rightarrow$	$\rightarrow$	$\rightarrow$
B			$\boxed{\|}$	$\rightarrow$
C		$\boxed{\|}$		$\rightarrow$
D				

A comparison of the behavioral relations in Tables 1 and 2 show that $BP_{L_{his}} \neq BP_{L_{act}}$. The changed behavioral relations are marked with dotted boxes. The log behavioral profiles have changed when the model does not perform a "Delete", "Move", or "Insert" change operation. It is assumed that the occurrence of hidden transitions affects the actual log behavioral relations. Therefore, it is necessary to investigate the behavioral changes of the model with hidden transitions using complete event logs.

3. Preliminaries of Models and Methods

We use event logs as our formal grounding. A class of Petri nets and a Petri net system is used for process modeling and analysis. This paper refers to basic definitions [23] of Petri nets, Petri net systems, and behavioral profiles. This section recalls some related concepts, whereas others are omitted for brevity.

Definition 1. *(Event Log* [24]*) Let T be the set of activities, where $\sigma \in T^*$ is an event trace. Let $L \in B(T^*)$ be an event log, where $B(T^*)$ is the set of event traces.*

Definition 2. *(Log Behavioral Profile (BP)* [25]*) Let $L_p = n_1, \cdots n_m$ be a log of a process model $P = (A, a_i, a_o, C, F, T)$. A pair $(x, y) \in (A_L \times A_L)$ has at most one of the following relations:*

(1) The strict order relation $\rightarrow_L$, if $x \succ_L y, y \succ_L x$;

(2) The interleaving order relation $||_L$, if $x \succ_L y, y \succ_L x$.

The set of relations is written as $BP_L = \{\rightarrow_L, \|_L\}$.

Definition 3. *(Log Minimum Successor Relation (MSR)* [26]*) Let $\sigma \in L$ be a log trace and let $x \in \sigma, y \in \sigma$ be the transitions, and $k \in N$. The log minimum successor relation is defined by: $x \succ_k^\sigma y \Leftrightarrow \exists 1 \leq i \leq |\sigma| [\sigma(i) = x \wedge \sigma(i+k) = y]$, and the log minimum successor relation is abbreviated as MSR. Therefore, $MSR_{(x,y)} = k$ in log L.*

In Definition 3, if transitions x and y are contained in the trace σ, such as $\sigma = < xx_1x_2x_3y >$, it can be obtained that $\sigma(1) = x \wedge \sigma(1+4) = y$. The log minimum successor relation of (x, y) is denoted as $MSR_{(x,y)} = 4$.

4. Change-Mining Methods for Models with Hidden Transitions Based on Complete Logs

Before introducing the behavioral change-mining method in detail, we first provide the classification rules and the types of hidden transition.

4.1. Classification Rules of Hidden Transitions

Let $N = (P, T, F)$ be a labelled Petri net, where $\tau_i, i \in \{1, 2, \cdots n\}$ denotes the hidden transition, $A \cup \tau_i$ denotes the set of all activities, and it holds that $t_i \in A$.

Rule 1. *If both* $^\bullet\tau_1 = t_1{}^\bullet = 1$ *and* $\tau_1{}^\bullet = {}^\bullet t_2 = 1$ *are satisfied, then* τ_1 *is defined as a strict hidden transition. Figure 1 gives a demonstration for Rule 1.*

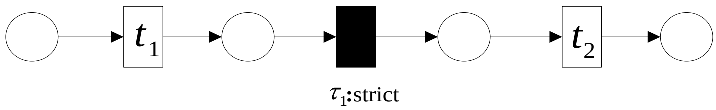

Figure 1. A case demonstration for strict hidden transition.

Rule 2. *If both* $^\bullet\tau_2 = 1$ *and* $\tau_2{}^\bullet > 1$ *hold, then* τ_2 *is defined as an and-split hidden transition. If both* $^\bullet\tau_3 > 1$ *and* $\tau_3{}^\bullet = 1$ *hold, then* τ_3 *is defined as an and-join hidden transition. Figure 2 gives a demonstration for Rule 2.*

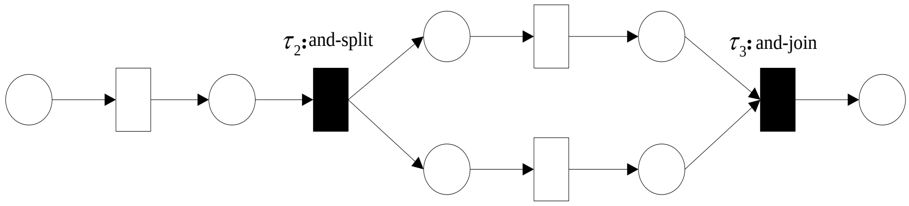

Figure 2. A case demonstration for the and-split hidden transition and and-join hidden transition.

Rule 3. *If* $^\bullet\tau_4 \cap t_3{}^\bullet \neq \varnothing$, $^\bullet\tau_4 \cap {}^\bullet t_4 = 1$ *and* $\tau_4{}^\bullet \cap t_4{}^\bullet = 1$ *are all satisfied, then* τ_4 *is defined as a skip hidden transition. Figure 3 gives a demonstration for Rule 3.*

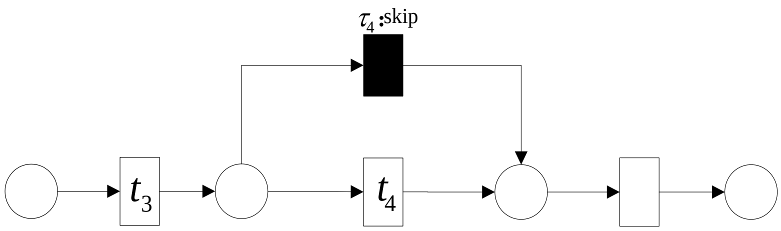

Figure 3. A case demonstration for the skip hidden transition.

Rule 4. *If both* $t_5{}^\bullet \cap \tau_5{}^\bullet \neq \varnothing$ *and* $^\bullet\tau_5 \cap {}^\bullet t_8 \neq \varnothing$ *are satisfied, then* τ_5 *is defined as a loop hidden transition. Figure 4 gives a demonstration for Rule 4.*

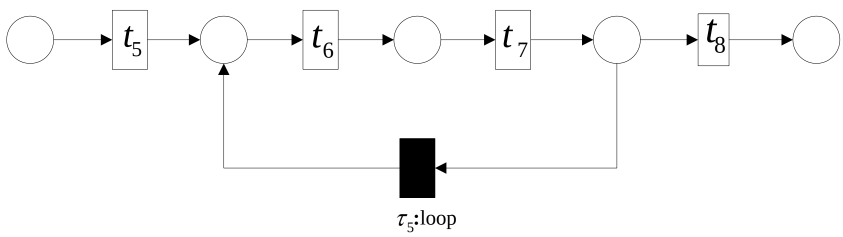

Figure 4. A case demonstration for the loop hidden transition.

Rule 5. *If both* $t_{11} + t_{14}$ *and* $t_{11}{}^\bullet = {}^\bullet\tau_6$, $\tau_6{}^\bullet = {}^\bullet t_{14}$ *are satisfied, then* τ_6 *is defined as a switch hidden transition. Figure 5 gives a demonstration for Rule 5.*

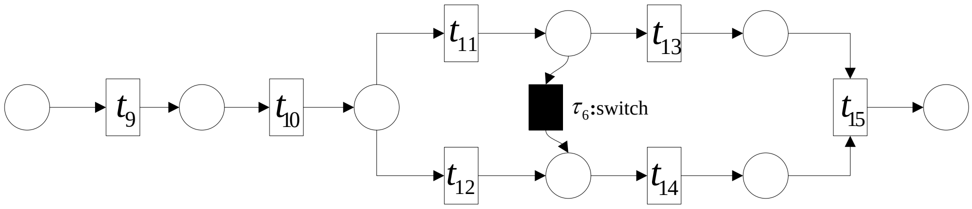

Figure 5. A case demonstration for the switch hidden transition.

4.2. Constructing Change-Mining Methods

This section mainly investigates the behavioral changes of models with hidden transitions under complete event logs. Previous research in the field of change mining has mainly focused on changes in model behavior [6–9] when the original system models are known. Additionally, few studies have concentrated on mining model changes based on incomplete event logs [11]. On the one hand, previous studies did not consider the effects of hidden transitions on model changes. On the other hand, hidden transitions are difficult to mine because they do not appear in any event trace. In the following, the related concepts of the log-weighted coefficient and the behavior-relation-change-mining algorithm are proposed.

Definition 4. *(Log-Weighted Coefficient) Let L be an event log and let $t_i \in L(i = 1, 2, 3 \ldots)$ be a transition. A coefficient is assigned to the arcs (drawn as directed edges) connecting transitions with others. This coefficient is called the log-weighted coefficient and denoted r.*

Example 1. *(Interpretation of the log-weighted coefficient) We first provide a workflow log $L =< \sigma_1, \sigma_2, \sigma_3, \sigma_4 >$, which contains $\sigma_1 =< ABHI >$, $\sigma_2 =< ADGECI >$, $\sigma_3 =< ADEGCI >$, and $\sigma 4 =< AGDECI >$. Transition activities (drawn as rectangles) are taken as the start and end points of each arc, and each arc is labelled with a coefficient (as shown in Figure 6), the log-weighted coefficient. It can be seen that (G, D) and (G, E) have the relations $G \succ_L D, D \succ_L G$ and $G \succ_L E, E \succ_L G$ according to $\sigma_2, \sigma_3, \sigma_4$. These relations are written as $G \parallel D$ and $G \parallel E$, respectively. Therefore, there are behavioral relations $G \underset{a}{\overset{a}{\rightleftarrows}} D$ $G \underset{b}{\overset{b}{\rightleftarrows}} E$ (where a, b are the weighted coefficients of arc G, D and G, E respectively). It is clear that $a = a$ and $b = b$ hold, and the relations of (G, D) and (G, E) are called a strict order relation and inverse strict order relation such that the values $\xrightarrow{a}$ and $\xleftarrow{a}$ offset each other. The following does not consider a pair of transitions with an interleaving order a behavioral relation.*

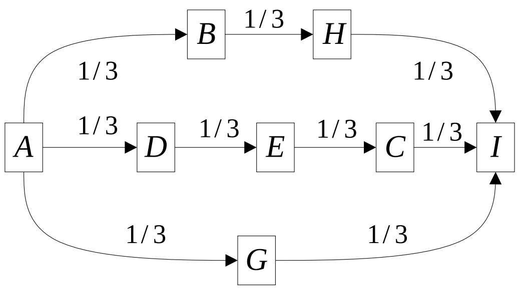

Figure 6. Illustration of the log-weighted coefficient.

Figure 6 gives a demonstration of Definition 4. The arcs of $A \to B \to H \to I$ are generated by the trace σ_1, and each arc is labelled with a weighted coefficient of 1/3. Specifically, there are three arcs connected to transition A, namely arc AB, arc AD, and arc AG, whose coefficients are $r_{AB} = \frac{1}{3}$, $r_{AD} = \frac{1}{3}$, and $r_{AG} = \frac{1}{3}$.

Definition 5. *(Rules for Determining the Log-Weighted Coefficient) For a complete event log L, if the weighted coefficients are given for each arc, the weighted coefficients of the event logs are uniquely determined. The log-weighted coefficients need to be determined according to the two following rules.*

Rule 6. *The coefficients of the output arcs are equal.*

In a complete event log, the log-weighted coefficients of the strict order structures generated by single transition nodes are equal. Generally, the value of the output arc coefficient of the transition is 1. If the transition has n output arcs, then the coefficients of these output arcs are $r_1 = r_2 = r_3 = \cdots = r_n = \frac{1}{n}$.

Rule 7. *The sum of the coefficients of the arcs is conserved.*

For any initial transition, the weighted coefficients of the output arcs are equal to the weighted coefficients of the input arcs of the final transition. Suppose there are n output arcs with coefficients $r_1 r_2 \cdots r_n$ and m input arcs with coefficients $b_1 b_2 \cdots b_m$ that satisfy $\sum_{i=1}^{n} a_i = \sum_{j=1}^{m} b_j$.

In the following, Algorithm 1 details the procedures of hidden transition change mining. The purpose of Algorithm 1 is to mine different types of hidden transitions and changed behavioral relations. To make it more readable, Algorithm 1 is divided into five phases. Taking the second phase as an example, we need to first determine whether the log L_{his} and L_{act} are equal. If L_{his} is not equal to L_{act}, BP, MSR, and r of the logs must be calculated (Definitions 2–4). If there is a pair of transition activity (a_i, b_j) satisfying $BP_{L_{his}}(a_i, b_j) = \varnothing$ and $BP_{L_{act}}(a_i, b_j) = \{\|_{L_{act}}\}$ under the condition of $MSR_{L_{act}} \neq MSR_{L_{his}}$, $r_{L_{act}} \neq r_{L_{his}}$, this leads to the conclusion that the hidden transition is and-split and the changed behavioral relation is $R_{ch} = BP_{L_{act}} - BP_{L_{his}} = BP_{L_{act}}(a_i, b_j) = \{\|_{L_{act}}\}$.

Algorithm 1 Behavior-Relation-Change-Mining Algorithm Considering Hidden Transitions (BRCM algorithm)

Input: Original complete event log L_{his}, actual complete event log L_{act}
Output: Changed behavioral relations R_{ch}
First, calculate the log behavioral profiles, denoted $BP_{L_{his}}$ and $BP_{L_{act}}$, the minimum successor relations, denoted $MSR_{L_{his}}$ and $MSR_{L_{act}}$, and the log-weighted coefficients, denoted $r_{L_{his}}$ and $r_{L_{act}}$.

Mining strict hidden transitions:
1 : Judge the equivalence of L_{his} and L_{act}.
2 : *If* $L_{act} = L_{his} \wedge \sigma_i \in L_{act} \wedge \sigma_j \in L_{his} (i = j = 1)$ hold,
3 : *then* it can be deduced $BP_{L_{act}} = BP_{L_{his}} \wedge MSR_{L_{act}} = MSR_{L_{his}} \wedge r_{L_{act}} = r_{L_{his}}$ hold,
4 : *return* $R_{ch} = \varnothing, \sigma_1 = \sigma_2$, and the hidden transition is a strict transition.

Mining and-split hidden transitions:
5 : Suppose that a set of transitions are mapped to $Y_{L_{his}}$ and $Y_{L_{act}}$.
6 : $If\ Y_{L_{his}} = Y_{L_{act}} \wedge L_{act} \neq L_{his} \wedge L_{act} \not\subset L_{his} \wedge L_{his} \not\subset L_{act} \wedge MSR_{L_{act}} \neq MSR_{L_{his}} \wedge r_{L_{act}} \neq r_{L_{his}}$ hold,
7 : *else if* $\exists a_i, b_j \in Y_{L_{his}}, i, j \in N$, because of $Y_{L_{his}} = Y_{L_{act}}$. Similarly, $\exists a_i, b_j \in Y_{L_{act}}, i, j \in N$. This. will not be described subsequently. Additionally, $BP_{L_{his}}\left(a_i, b_j\right) = \varnothing$ and $BP_{L_{act}}\left(a_i, b_j\right) = \{\|_{L_{act}}\}$.
8 : *Return* $R_{ch} = BP_{L_{act}} - BP_{L_{his}}$. The hidden transition is an and-split transition. Using the same method, we deduce the and-join hidden transition and the changed behavioral relations R_{ch}.

Mining skip hidden transitions:
9 : $If\ Y_{L_{his}} = Y_{L_{act}} \wedge L_{act} \neq L_{his} \wedge MSR_{L_{act}} \neq MSR_{L_{his}} \wedge r_{L_{act}} \neq r_{L_{his}}$ hold,
10 : *else if* $L_{act} \cap L_{his} = L_{his} \neq L_{act}$,
11 : *return* $BP_{L_{his}} \subseteq BP_{L_{act}}$.
12 : *If* $\exists a_i, b_j \in Y_{L_{his}}, i, j \in N, BP_{L_{his}}\left(a_i, b_j\right) = \{\rightarrow_{L_{his}}\} = BP_{L_{act}}\left(a_i, b_j\right), MSR_{L_{his}}(a_i, b_j) = n,$ and $MSR_{L_{act}}(a_i, b_j) < n$ hold,
13 : *else if* $\exists c_i, d_j, e_m \in Y_{L_{his}}, i, j, m \in N$. According to the original event log the value of, the weighted coefficient is $r_{L_{his}}\left(c_i, d_j\right) = 1$. However, the value of the weighted coefficient is $r_{L_{act}}\left(c_i, d_j\right) = r_{L_{act}}(c_i, e_m) = \frac{1}{2}$ in the actual log.

Algorithm 1 *Cont.*

$Return\ R_{ch} = BP_{L_{act}} - BP_{L_{his}}$. The hidden transition is a skip transition.

Mining loop hidden transitions:
$If\ Y_{L_{his}} = Y_{L_{act}} \wedge L_{act} \neq L_{his} \wedge MSR_{L_{act}} \neq MSR_{L_{his}} \wedge r_{L_{act}} \neq r_{L_{his}}$ hold,
$else\ if\ \exists a_i\ or\ b_j \in Y_{L_{his}}, i, j, m \in N.\ BP_{L_{his}}(a_i, a_i) = BP_{L_{his}}\left(b_j, b_j\right) = \varnothing,\ BP_{L_{act}}(a_i, a_i) \neq \varnothing,$
$BP_{L_{act}}\left(b_j, b_j\right) \neq \varnothing, MSR_{L_{his}}(a_i, a_i) = MSR_{L_{his}}\left(b_j, b_j\right) = \varnothing, MSR_{L_{act}}(a_i, a_i) \neq \varnothing,$
and $MSR_{L_{act}}\left(b_j, b_j\right) \neq \varnothing$ hold.
There exist $a_i, b_j \in Y_{L_{his}}, i, j \in N$ satisfying one of the following behavioral relations :
$(1) BP_{L_{his}}(a_i, a_i) = \varnothing,\ BP_{L_{act}}(a_i, a_i) = \{\rightarrow\};\ (2) a_i \succ_{L_{his}} b_j, b_j \succ_{L_{his}} a_i, a_i \succ_{L_{act}} b_j, b_j \succ_{L_{act}}$
$a_i;\ (3)\ a_i \succ_{L_{his}} a_i, b_j \succ_{L_{his}} b_j,\ a_i \succ_{L_{act}} a_i, b_j \succ_{L_{act}} b_j.$
$Return R_{ch} = BP_{L_{act}} - BP_{L_{his}}$, and the hidden transition is a loop transition.

Mining switch hidden transitions:
$If\ Y_{L_{his}} = Y_{L_{act}} \wedge L_{act} \neq L_{his} \wedge L_{act} \cap L_{his} \neq \varnothing \wedge MSR_{L_{act}} \neq MSR_{L_{his}} \wedge r_{L_{act}} \neq r_{L_{his}}$ hold,
$else\ if$ there exists $\left(a_i, b_j\right) \in Y_{L_{his}}, i, j \in N$ satisfying $BP_{L_{his}}\left(a_i, b_j\right) = \varnothing,$
$return\ MSR_{L_{his}}\left(a_i, b_j\right) = \varnothing$ and $r_{L_{his}}\left(a_i, b_j\right) = \varnothing.$
$else\ if\ BP_{L_{act}}\left(a_i, b_j\right) = \{\rightarrow_{L_{act}}\} \neq \varnothing \wedge MSR_{L_{act}}\left(a_i, b_j\right) = 1 \wedge r_{L_{act}}\left(a_i, b_j\right) > 0,$
$return\ R_{ch} = BP_{L_{act}} - BP_{L_{his}}$. The hidden transition is a switch transition.

4.3. Examples

We choose the specific event logs with hidden transitions for analysis in this section. Five groups of complete logs are selected for investigation, and each group of logs contains at least one execution trace. Table 3 provides the relevant information on the trace for the first group of event logs.

Table 3. Original event logs and actual logs in group 1.

Symbol	Trace
σ_1	$L_{his} =<< ABC >>$
σ_2	$L_{act} =<< ABC >>$

Table 3 shows that the original event log is $L_{his} = \{< ABC >\}$ and the actual log is $L_{act} = \{< ABC >\}$, where $L_{his} = L_{act}$, $MSR_{L_{act}} = MSR_{L_{his}}$ and $r_{L_{act}} = r_{L_{his}}$ hold. In terms of Algorithm 1, we conclude that the actual log L_{act} contains a strict hidden transition and the changed behavioral relation is $R_{ch} = \varnothing$.

Table 4 gives the relevant information of the trace for the second group of event logs.

Table 4. Original event logs and actual logs in group 2.

Symbol	Trace
σ_3	$L_{his} =<< ABCF >>$
σ_4	$L_{his} =<< ADEF >>$
σ_5	$L_{act} =<< ABDCEF >>$
σ_6	$L_{act} =<< ABDCEF >>$
σ_7	$L_{act} =<< ADBCEF >>$
σ_8	$L_{act} =<< ADBECF >>$

Table 4 shows the original event log $L_{his} = \{< ABCF >, < ADEF >\}$ and the actual event log $L_{act} = \{< ABDCEF >, < ABDECF >, < ADBCEF >, < ADBECF >\}$. We calculate log behavioral profiles $BP_{L_{his}}$ and $BP_{L_{act}}$ according to Definition 2. The log minimum successor relations calculated using Definition 3 are given in Tables 5 and 6.

Table 5. Behavioral profiles ($BP_{L_{his}}$) and log minimum successor relations ($MSR_{L_{his}})_{(\text{group 2})}$.

	A	B	C	D	E	F
A		→(1)	→(2)	→(1)	→(2)	→(3)
B			→(1)			→(2)
C						→(1)
D					→(1)	→(2)
E						→(1)
F						

Table 6. Behavioral profiles ($BP_{L_{act}}$) and log minimum successor relations ($MSR_{L_{act}})_{(\text{group 2})}$.

	A	B	C	D	E	F
A		→(1)	→(3)	→(1)	→(3)	→(5)
B			→(1)	‖(1)	→(1)	→(4)
C					‖(1)	→(3)
D		‖(1)	→(1)		→(1)	→(2)
E			‖(1)			→(1)
F						

Table 6 shows the minimum successor relations of transition activities. For example, the behavioral relation between transition activities A and B is a strict order relation, and the value of minimum successor relation is 1, marked as →(1). Figures 7 and 8 present two case descriptions of log-weighted coefficients for L_{his} and L_{act} according to Definition 4.

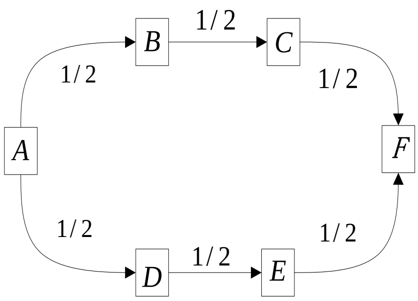

Figure 7. Case description for L_{his}.

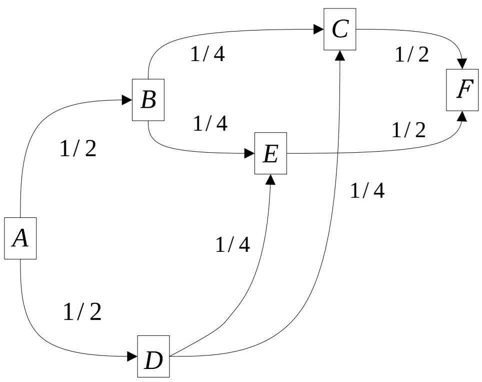

Figure 8. Case description for L_{act}.

A comparison of Tables 5 and 6 clearly shows that there are two pairs of transition activities (B, D) and (C, E), satisfying $BP_{L_{his}}(B, D) = \varnothing$, $BP_{L_{act}}(B, D) = \{\|\}$, $BP_{L_{his}}(C, E) = \varnothing$, and $BP_{L_{act}}(C, E) = \{\|\}$. $MSR_{L_{act}} \neq MSR_{L_{his}}$ and $r_{L_{act}} \neq r_{L_{his}}$ are under the condition that $Y_{L_{his}} = Y_{L_{act}}$, where the actual log L_{act} contains an and-split hidden transition and the changed behavioral relations are $R_{ch} = BP_{L_{act}} - BP_{L_{his}} = BP_{(B,D)}(\|_{L_{act}}) + BP_{(B,E)}(\rightarrow_{L_{act}}) + BP_{(C,E)}(\|_{L_{act}}) + BP_{(D,C)}(\rightarrow_{L_{act}})$.

Table 7 shows the relevant information of the trace for the third group of event logs.

Table 7. Original event logs and actual logs in group 3.

Symbol	Trace
σ_9	$L_{his} =<< ABC >>$
σ_{10}	$L_{act} =<< ABC >>$
σ_{11}	$L_{act} =<< AC >>$

Tables 8 and 9 show the log behavioral profiles and log minimum successor relations calculated using Definitions 2 and 3, respectively.

Table 8. Behavioral profiles ($BP_{L_{his}}$) and log minimum successor relations ($MSR_{L_{his}})_{(\text{group 3})}$.

	A	B	C
A		$\rightarrow$(1)	$\rightarrow$(2)
B			$\rightarrow$(1)
C			

Table 9. Behavioral profiles ($BP_{L_{act}}$) and log minimum successor relations ($MSR_{L_{act}})_{(\text{group 3})}$.

	A	B	C
A		$\rightarrow$(1)	$\rightarrow$(1)
B			$\rightarrow$(1)
C			

We see that the original logs and actual logs satisfy $L_{act} \cap L_{his} = L_{his}$. According to Definition 5, the log-weighted coefficients are calculated as $r_{L_{his}}(A, B) = r_{L_{his}}(B, C) = 1$ and $r_{L_{act}}(A, B) = r_{L_{act}}(A, C) = r_{L_{act}}(B, C) = \frac{1}{2}$. It is inferred that the actual log L_{act} contains a skip hidden transition and the changed behavioral relation is $R_{ch} = BP_{L_{act}} - BP_{L_{his}} = \varnothing$. Meanwhile, the log minimum successor relations changed from $MSR_{L_{his}}(A, C) = 2$ to $MSR_{L_{act}}(A, C) = 1$.

Tables 10 and 11 present the log instances of groups 4 and 5, respectively. $\dot{B}\dot{C}$ indicates that the transition activity BC is enabled in an infinite loop in σ_{14}.

Table 10. Original event logs and actual logs in group 4.

Symbol	Trace
σ_{12}	$L_{his} =<< ABCD >>$
σ_{13}	$L_{act} =<< ABCD >>$
σ_{14}	$L_{act} =<< ABCBCBC\dot{B}\dot{C} >>$

Table 11. Original event logs and actual logs in group 5.

Symbol	Trace
σ_{15}	$L_{his} =<< ABCDG >>$
σ_{16}	$L_{his} =<< ABEFG >>$
σ_{17}	$L_{act} =<< ABCDG >>$
σ_{18}	$L_{act} =<< ABEFG >>$
σ_{19}	$L_{act} =<< ABCFG >>$

Tables 12 and 13 show the calculated log behavioral profiles and log minimum successor relations of group 4, respectively.

Table 12. Behavioral profiles ($BP_{L_{his}}$) and log minimum successor relations ($MSR_{L_{his}}$)$_{\text{(group 4)}}$.

	A	B	C	D
A		→(1)	→(2)	→(3)
B			→(1)	→(2)
C				→(1)
D				

Table 13. Behavioral profiles ($BP_{L_{act}}$) and log minimum successor relations ($MSR_{L_{act}}$)$_{\text{(group 4)}}$.

	A	B	C	D
A		→(1)	→(2)	→(3)
B		→(2)	||(1)	→(2)
C		||(1)	→(2)	→(1)
D				

It is inferred that $Y_{L_{his}} = Y_{L_{act}}$, $L_{act} \neq L_{his}$, $MSR_{L_{act}} \neq MSR_{L_{his}}$, and $r_{L_{act}} \neq r_{L_{his}}$ hold. Moreover, the transition activity B, C satisfies $BP_{L_{his}}(B,B) = BP_{L_{his}}(C,C) = \varnothing$, $BP_{L_{act}}(B,B) \neq \varnothing$, $BP_{L_{act}}(C,C) \neq \varnothing$, $MSR_{L_{his}}(B,B) = MSR_{L_{his}}(C,C) = \varnothing$, $MSR_{L_{act}}(B,B) \neq \varnothing$, and $MSR_{L_{act}}(C,C) \neq \varnothing$. The pair transition activity (B,C) satisfies ① $B \succ_{L_{act}} C, C \succ_{L_{act}} B$, $B \succ_{L_{his}} C, C \succ_{L_{his}} B$, ② $B \succ_{L_{his}} B, C \succ_{L_{his}} C$, and $B \succ_{L_{act}} B, C \succ_{L_{act}} C$.

Additionally, Tables 12 and 13 show that the changed behavioral relations are $R_{ch} = BP_{L_{act}} - BP_{L_{his}} = BP_{(B,B)}(\rightarrow_{L_{act}}) + BP_{(B,C)}(\|_{L_{act}}) + BP_{(C,C)}(\rightarrow_{L_{act}})$ in the actual logs and contain the loop hidden transition.

Tables 14 and 15 show the calculated log behavioral profiles and log minimum successor relations of group 5, respectively.

Table 14. Behavioral profiles ($BP_{L_{his}}$) and log minimum successor relations ($MSR_{L_{his}}$)$_{\text{(group 5)}}$.

	A	B	C	D	E	F	G
A		→(1)	→(2)	→(3)	→(2)	→(3)	→(4)
B			→(1)	→(2)	→(1)	→(2)	→(3)
C				→(1)			→(2)
D							→(1)
E						→(1)	→(2)
F							→(1)
G							

Table 15. Behavioral profiles ($BP_{L_{act}}$) and log minimum successor relations ($MSR_{L_{act}}$)$_{\text{(group 5)}}$.

	A	B	C	D	E	F	G
A		→(1)	→(2)	→(3)	→(2)	→(3)	→(4)
B			→(1)	→(2)	→(1)	→(2)	→(3)
C				→(1)		→(1)	→(2)
D							→(1)
E						→(1)	→(2)
F							→(1)
G							

For the logs of group 5, $Y_{L_{his}} = Y_{L_{act}}$, $L_{act} \neq L_{his} \wedge L_{act} \cap L_{his} = L_{his} \neq \varnothing$, and $MSR_{L_{act}} \neq MSR_{L_{his}}$. According to Definition 4, the log-weighted coefficients are calculated as $r_{L_{act}} \neq r_{L_{his}}$. A comparison of Tables 14 and 15 clearly shows that $BP_{L_{his}}(C,F) = \varnothing$ is satisfied.

However, in the actual log L_{act}, $BP_{L_{act}}(C,F) = \{\rightarrow_{L_{act}}\} \neq \varnothing$ and $MSR_{L_{act}}(C,F) = 1$ $\wedge r_{L_{act}}(C,F) > 0$ are satisfied. The changed relation is written as $R_{ch} = BP_{L_{act}} - BP_{L_{his}} = BP_{(C,F)}(\rightarrow_{L_{act}})$ and the changed transition, called a switch hidden transition, is mined in the actual log.

In this section, we discussed several behavioral-change-mining methods. The behavioral relation between the strict hidden transition and other transitions is usually a strict order relation, such that the strict hidden transition does not affect the model behavior. Meanwhile, the effects of four types of hidden transitions, namely the and-split, skip, loop, and switch transitions, on the model behavior can be divided into four categories:

(1) Strict order relation $\{\rightarrow\} \Rightarrow$ interleaving order relation $\{\|\}$;
(2) Interleaving order relation $\{\|\} \Rightarrow$ strict order relation $(\rightarrow)$;
(3) $BP_{(a_1,a_2)} = \varnothing \Rightarrow BP_{(a_1,a_2)} = \{\|\}$ or $BP_{(a_1,a_2)} = \{\|\} \Rightarrow BP_{(a_1,a_2)} = \varnothing$;
(4) $BP_{(a_1,a_2)} = \varnothing \Rightarrow BP_{(a_1,a_2)} = \{\rightarrow\}$ or $BP_{(a_1,a_2)} = \{\rightarrow\} \Rightarrow BP_{(a_1,a_2)} = \varnothing$.

5. Experimental Evaluation

The computing resource used in the experiments was an Intel Core I5 eight-generation central processing unit with 8 GB RAM and a Windows 10 64-bit operating system. The event logs used in the experiment were typical artificial structured control flow event logs, namely Artificial Structured Control Flow. Figure 9 presents the results obtained using the Inductive Miner method in the ProM process mining tool to model the artificial structured control flow event logs.

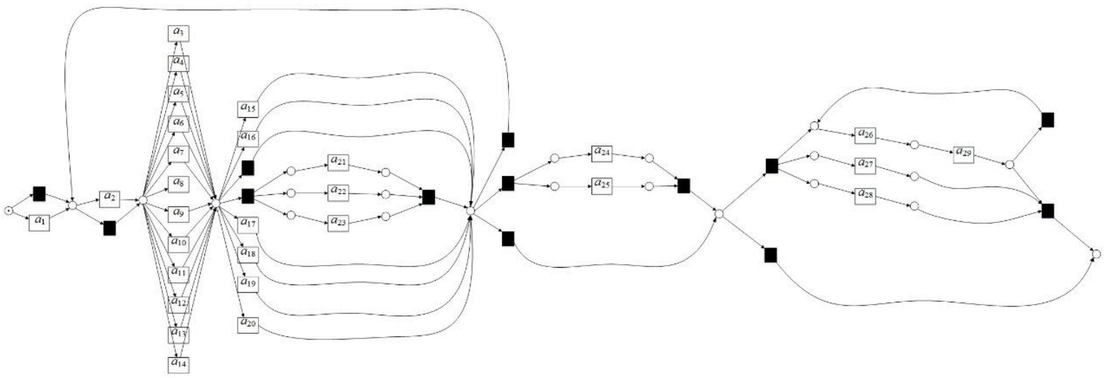

Figure 9. Model mining results obtained using ProM (Inductive Miner).

Figure 9 shows that the number of typical structured control flow event logs is large, the behavioral relations are relatively complex, and the workload of studying the overall behavior structure of the model is heavy. A representative local model is thus selected as the research object to further analyze and verify the effectiveness of the algorithm proposed in this paper. Figure 10 is a partial view of the ProM simulation diagram mined from the artificial structured control flow event logs. Table 16 presents model logs obtained using the online simulation tool CPN Tools, denoted L_{his}.

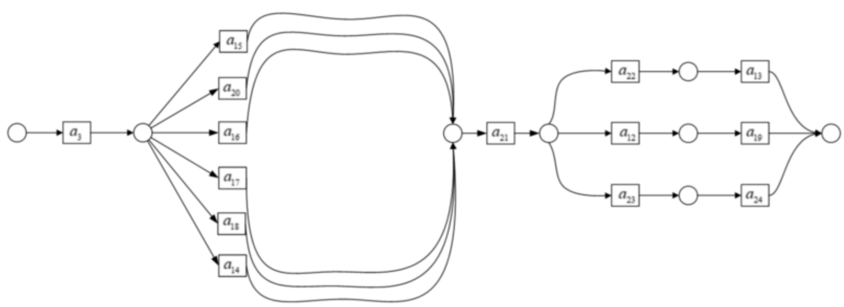

Figure 10. Partial view of the ProM simulation diagram.

Table 16. Original event logs L_{his} obtained by CPN Tools.

Symbol	Trace	Symbol	Trace
σ_1	$L_{his} = <<a_3a_{15}a_{21}a_{22}a_{13}>>$	σ_{10}	$L_{his} = <<a_3a_{17}a_{21}a_{22}a_{13}>>$
σ_2	$L_{his} = <<a_3a_{15}a_{21}a_{12}a_{19}>>$	σ_{11}	$L_{his} = <<a_3a_{17}a_{21}a_{12}a_{19}>>$
σ_3	$L_{his} = <<a_3a_{15}a_{21}a_{23}a_{24}>>$	σ_{12}	$L_{his} = <<a_3a_{17}a_{21}a_{23}a_{24}>>$
σ_4	$L_{his} = <<a_3a_{20}a_{21}a_{22}a_{13}>>$	σ_{13}	$L_{his} = <<a_3a_{18}a_{21}a_{22}a_{13}>>$
σ_5	$L_{his} = <<a_3a_{20}a_{21}a_{12}a_{19}>>$	σ_{14}	$L_{his} = <<a_3a_{18}a_{21}a_{12}a_{19}>>$
σ_6	$L_{his} = <<a_3a_{20}a_{21}a_{23}a_{24}>>$	σ_{15}	$L_{his} = <<a_3a_{18}a_{21}a_{23}a_{24}>>$
σ_7	$L_{his} = <<a_3a_{16}a_{21}a_{22}a_{13}>>$	σ_{16}	$L_{his} = <<a_3a_{14}a_{21}a_{22}a_{13}>>$
σ_8	$L_{his} = <<a_3a_{16}a_{21}a_{12}a_{19}>>$	σ_{17}	$L_{his} = <<a_3a_{14}a_{21}a_{12}a_{19}>>$
σ_9	$L_{his} = <<a_3a_{16}a_{21}a_{23}a_{24}>>$	σ_{18}	$L_{his} = <<a_3a_{14}a_{21}a_{23}a_{24}>>$

Table 17 presents the behavioral profiles and log minimum successor relations of L_{his} calculated using Definitions 2 and 3.

Table 17. $BP_{L_{his}}$ and $MSR_{L_{his}}$ corresponding to the logs in Table 16.

	a_3	a_{12}	a_{13}	a_{14}	a_{15}	a_{16}	a_{17}	a_{18}	a_{19}	a_{20}	a_{21}	a_{22}	a_{23}	a_{24}
a_3		$\rightarrow$(3)	$\rightarrow$(4)	$\rightarrow$(1)	$\rightarrow$(1)	$\rightarrow$(1)	$\rightarrow$(1)	$\rightarrow$(1)	$\rightarrow$(4)	$\rightarrow$(1)	$\rightarrow$(2)	$\rightarrow$(3)	$\rightarrow$(3)	$\rightarrow$(4)
a_{12}									$\rightarrow$(1)					
a_{13}														
a_{14}		$\rightarrow$(2)	$\rightarrow$(3)						$\rightarrow$(3)		$\rightarrow$(1)	$\rightarrow$(2)	$\rightarrow$(2)	$\rightarrow$(3)
a_{15}		$\rightarrow$(2)	$\rightarrow$(3)						$\rightarrow$(3)		$\rightarrow$(1)	$\rightarrow$(2)	$\rightarrow$(2)	$\rightarrow$(3)
a_{16}		$\rightarrow$(2)	$\rightarrow$(3)						$\rightarrow$(3)		$\rightarrow$(1)	$\rightarrow$(2)	$\rightarrow$(2)	$\rightarrow$(3)
a_{17}		$\rightarrow$(2)	$\rightarrow$(3)						$\rightarrow$(3)		$\rightarrow$(1)	$\rightarrow$(2)	$\rightarrow$(2)	$\rightarrow$(3)
a_{18}		$\rightarrow$(2)	$\rightarrow$(3)						$\rightarrow$(3)		$\rightarrow$(1)	$\rightarrow$(2)	$\rightarrow$(2)	$\rightarrow$(3)
a_{19}														
a_{20}		$\rightarrow$(2)	$\rightarrow$(3)						$\rightarrow$(3)		$\rightarrow$(1)	$\rightarrow$(2)	$\rightarrow$(2)	$\rightarrow$(3)
a_{21}		$\rightarrow$(1)	$\rightarrow$(2)						$\rightarrow$(2)			$\rightarrow$(1)	$\rightarrow$(1)	$\rightarrow$(2)
a_{22}			$\rightarrow$(1)											
a_{23}														$\rightarrow$(1)
a_{24}														

The actual logs that contain the hidden transitions are presented in Table 18, and the corresponding behavioral profile relations and log minimum successor relations are presented in Table 19.

Table 18. Actual event logs L_{act}.

Symbol	Trace	Symbol	Trace
σ_1	$L_{act} = <<a_3a_{15}a_{21}a_{22}a_{13}>>$	σ_{16}	$L_{act} = <<a_3a_{17}a_{16}a_{21}a_{22}a_{13}>>$
σ_2	$L_{act} = <<a_3a_{15}a_{21}a_{22}\dot{a}_{13}>>$	σ_{17}	$L_{act} = <<a_3a_{17}a_{16}a_{21}a_{22}\dot{a}_{13}>>$
σ_3	$L_{act} = <<a_3a_{15}a_{21}a_{12}a_{19}>>$	σ_{18}	$L_{act} = <<a_3a_{17}a_{16}a_{21}a_{12}a_{19}>>$
σ_4	$L_{act} = <<a_3a_{15}a_{21}a_{23}a_{24}>>$	σ_{19}	$L_{act} = <<a_3a_{17}a_{16}a_{21}a_{23}a_{24}>>$
σ_5	$L_{act} = <<a_3a_{15}a_{21}a_{12}a_{24}>>$	σ_{20}	$L_{act} = <<a_3a_{17}a_{16}a_{21}a_{12}a_{24}>>$
σ_6	$L_{act} = <<a_3a_{20}a_{21}a_{22}a_{13}>>$	σ_{21}	$L_{act} = <<a_3a_{18}a_{21}a_{22}a_{13}>>$
σ_7	$L_{act} = <<a_3a_{20}a_{21}a_{22}\dot{a}_{13}>>$	σ_{22}	$L_{act} = <<a_3a_{18}a_{21}a_{22}\dot{a}_{13}>>$
σ_8	$L_{act} = <<a_3a_{20}a_{21}a_{12}a_{19}>>$	σ_{23}	$L_{act} = <<a_3a_{18}a_{21}a_{12}a_{19}>>$
σ_9	$L_{act} = <<a_3a_{20}a_{21}a_{23}a_{24}>>$	σ_{24}	$L_{act} = <<a_3a_{18}a_{21}a_{23}a_{24}>>$
σ_{10}	$L_{act} = <<a_3a_{20}a_{21}a_{12}a_{24}>>$	σ_{25}	$L_{act} = <<a_3a_{18}a_{21}a_{12}a_{24}>>$
σ_{11}	$L_{act} = <<a_3a_{16}a_{17}a_{21}a_{22}a_{13}>>$	σ_{26}	$L_{act} = <<a_3a_{14}a_{21}a_{22}a_{13}>>$
σ_{12}	$L_{act} = <<a_3a_{16}a_{17}a_{21}a_{22}\dot{a}_{13}>>$	σ_{27}	$L_{act} = <<a_3a_{14}a_{21}a_{22}\dot{a}_{13}>>$
σ_{13}	$L_{act} = <<a_3a_{16}a_{17}a_{21}a_{12}a_{19}>>$	σ_{28}	$L_{act} = <<a_3a_{14}a_{21}a_{12}a_{19}>>$
σ_{14}	$L_{act} = <<a_3a_{16}a_{17}a_{21}a_{23}a_{24}>>$	σ_{29}	$L_{act} = <<a_3a_{14}a_{21}a_{23}a_{24}>>$
σ_{15}	$L_{act} = <<a_3a_{16}a_{17}a_{21}a_{12}a_{24}>>$	σ_{30}	$L_{act} = <<a_3a_{14}a_{21}a_{12}a_{24}>>$

It is first inferred that $Y_{L_{his}} = Y_{L_{act}}$, $L_{act} \neq L_{his} \wedge L_{act} \not\subset L_{his} \wedge L_{his} \not\subset L_{act}$, $MSR_{L_{act}} \neq MSR_{L_{his}}$, and $r_{L_{act}} \neq r_{L_{his}}$ hold. The following conclusions are then drawn from the operation and analysis of Algorithm 1.

(1) $a_{16}, a_{17} \in Y_{his}$ are present. Similarly, $a_{16}, a_{17} \in Y_{act}$ are present because $Y_{his} = Y_{act}$. Additionally, we obtain $BP_{L_{his}}(a_{16}, a_{17}) = \varnothing$ and $BP_{L_{act}}(a_{16}, a_{17}) = \{\|_{act}\}$. It is inferred from Algorithm 1 (5–8) that the hidden transition contained in the actual log is an

and-split transition, and the changed behavioral relation is $R_{ch} = BP_{L_{act}}(a_{16}, a_{17}) - BP_{L_{his}}(a_{16}, a_{17}) = \{\|\}$.

(2) The following are present: $a_{13} \in Y_{L_{his}}$, $BP_{L_{his}}(a_{13}, a_{13}) = \varnothing$, $BP_{L_{act}}(a_{13}, a_{13}) = \{\rightarrow\}$, $MSR_{L_{his}}(a_{13}, a_{13}) = \varnothing$, and $MSR_{L_{act}}(a_{13}, a_{13}) = 1 \neq \varnothing$. It is inferred from Algorithm 1 (15–18) that the hidden transition is a loop transition, and the changed behavioral relation is $R_{ch} = BP_{L_{act}}(a_{13}, a_{13}) - BP_{L_{his}}(a_{13}, a_{13}) = \{\rightarrow\}$.

(3) $(a_{12}, a_{24}) \in Y_{L_{his}}$ satisfies $BP_{L_{his}}(a_{12}, a_{24}) = \varnothing$ and $MSR_{L_{his}}(a_{12}, a_{24}) = \varnothing$ such that we obtain $r_{L_{his}}(a_{12}, a_{24}) = \varnothing$. However, $BP_{L_{act}}(a_{12}, a_{24}) = \{\rightarrow_{L_{act}}\} \neq \varnothing$ and $MSR_{L_{act}}(a_{12}, a_{24}) = 1$ hold. $r_{L_{act}}(a_{12}, a_{24}) = \frac{1}{6}$ is obtained from Definition 5. It is inferred from Algorithm 1 (19–23) that the hidden transition is a switch transition, and the changed behavioral relation is $R_{ch} = BP_{L_{act}}(a_{12}, a_{24}) - BP_{L_{his}}(a_{12}, a_{24}) = \{\rightarrow\}$.

Table 19. $BP_{L_{act}}$ and $MSR_{L_{act}}$ of L_{act}.

	a_3	a_{12}	a_{13}	a_{14}	a_{15}	a_{16}	a_{17}	a_{18}	a_{19}	a_{20}	a_{21}	a_{22}	a_{23}	a_{24}
a_3		→(3)	→(4)	→(1)	→(1)	→(1)	→(1)	→(1)	→(4)	→(1)	→(2)	→(3)	→(3)	→(4)
a_{12}									→(1)					→(1)
a_{13}			→(1)											
a_{14}		→(2)	→(3)						→(3)		→(1)	→(2)	→(2)	→(3)
a_{15}		→(2)	→(3)						→(3)		→(1)	→(2)	→(2)	→(3)
a_{16}		→(2)	→(3)				‖(1)		→(3)		→(1)	→(2)	→(2)	→(3)
a_{17}		→(2)	→(3)			‖(1)			→(3)		→(1)	→(2)	→(2)	→(3)
a_{18}		→(2)	→(3)						→(3)		→(1)	→(2)	→(2)	→(3)
a_{19}														
a_{20}		→(2)	→(3)						→(3)		→(1)	→(2)	→(2)	→(3)
a_{21}		→(1)	→(2)						→(2)			→(1)	→(1)	→(2)
a_{22}			→(1)											
a_{23}														→(1)
a_{24}														

To verify the correctness of the method, the artificial structured control flow event logs with hidden transitions are mined using ProM Tools, and the Petri net model shown in Figure 11 is obtained. The model shown in Figure 11 differs from that in Figure 10 due to the triggering of hidden transitions. As a more intuitive illustration, the model with hidden transitions in Figure 11 is drawn in Figure 12a, and the original model without hidden transitions is drawn in Figure 12b.

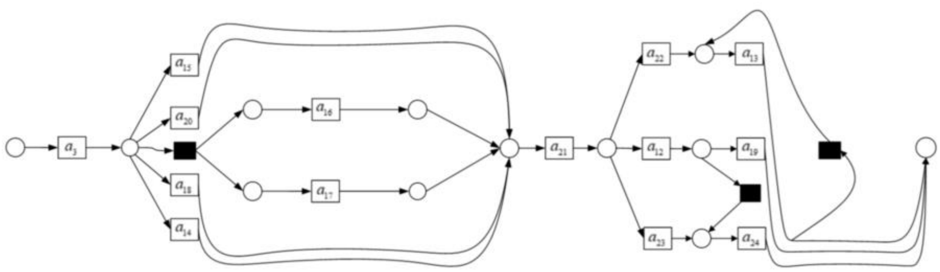

Figure 11. Model mined by ProM with hidden transitions.

It is seen that and-split, loop, and switch hidden transitions are enabled in the actual model. The simulation results are consistent with the calculation results of the method presented in this paper regarding the types of change transitions and the model behavioral relations, which further verifies the correctness of the method.

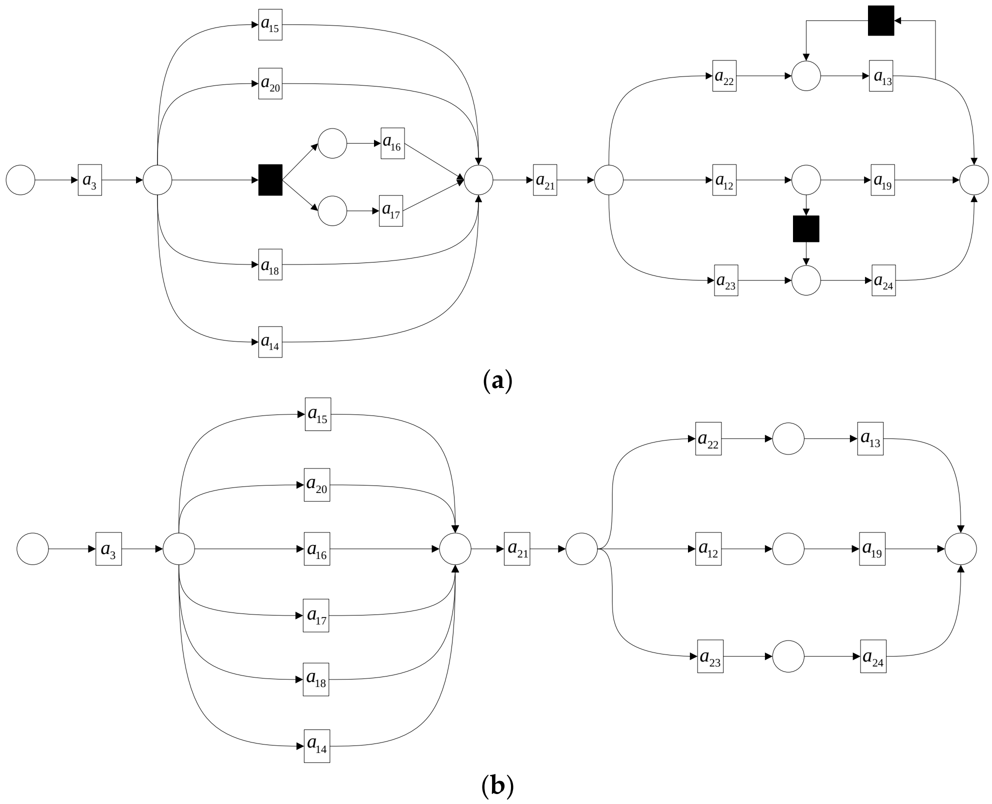

Figure 12. Comparisons of two types of models: (**a**) model with hidden transitions; (**b**) original model without hidden transitions.

6. Case Study

6.1. Basic Types of Disaster Chains

Research on disaster chains is still in its infancy, and further research in terms of risk assessment, disaster loss estimation, and chain-cutting disaster mitigation is necessary in conjunction with the adoption of other related technologies. The Petri net as a mathematical tool for modeling discrete and distributed systems has been widely used in risk assessment and disaster chain analysis in recent years. Additionally, the Petri net can model the disaster chains and express the triggering factors in the assessment of safety, reliability, and risk. Before conducting the actual disaster scenario analysis, four basic types of disaster chain are introduced from the perspective of disaster interrelationships, namely the causal disaster chain, concurrent disaster chain, exclusive disaster chain, and coupled disaster chain. These basic forms are shown in Figure 13.

Figure 13a shows the causal disaster chain: DSE_{chain1} triggers DSE_{chain2}. The sequential structure of the causal disaster chain is relatively simple and can help to distinguish between precursor disasters and subsequent disasters, such as $Pre(DSE_{chain2}) = DSE_{chain1}$ and $Sub(DSE_{chain1}) = DSE_{chain2}$. Figure 13b shows the concurrent disaster chain:DSE_{chain1} triggers DSE_{chain2} and DSE_{chain3}.The probabilities of subsequent disasters are equal, i.e., $r(DSE_{chain1}DSE_{chain2}) = r(DSE_{chain1}DSE_{chain3})$. Figure 13c shows the exclusive disaster chain, where the occurrence of disaster events A and B is exclusive, meaning that one disaster occurs and the other does not occur or diminishes. Figure 13d shows the coupled disaster chain where DSE_{chain1} and DSE_{chain2} together trigger DSE_{chain3}. The risk of a coupled disaster chain is higher than the risk posed by other disaster chains.

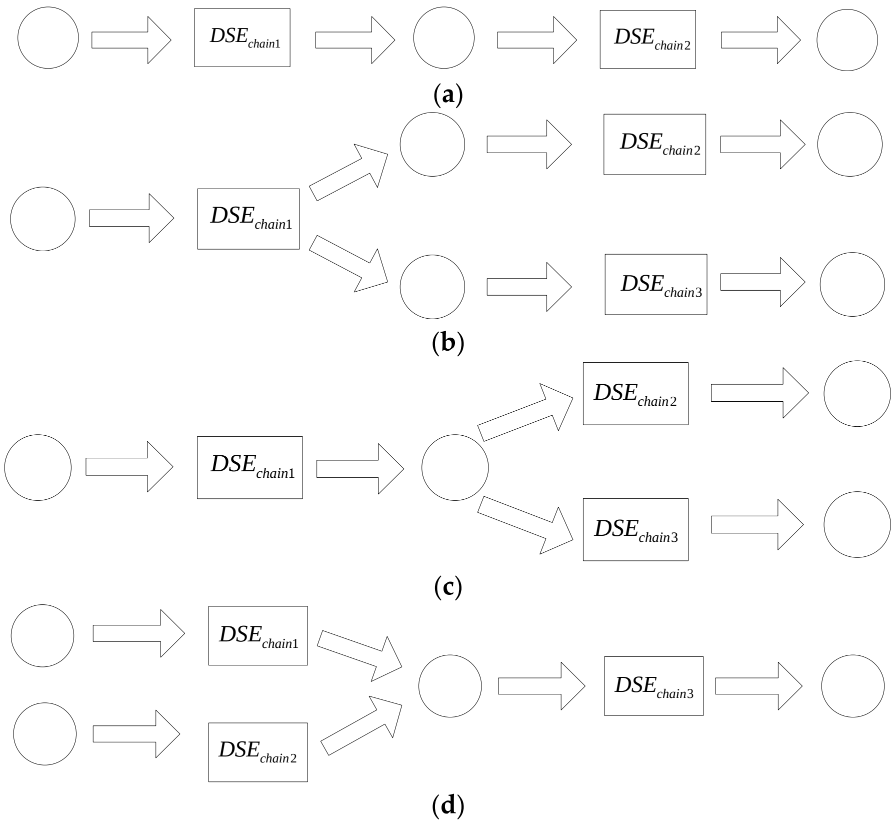

Figure 13. Basic types of disaster chain: (**a**) causal disaster chain; (**b**) concurrent disaster chain; (**c**) exclusive disaster chain; (**d**) coupled disaster chain.

6.2. Application of the Method

The dynamic evolution of a disaster chain is analyzed for the environmental pollution and ecological damage caused by a petrochemical leak, taking the 2018 petrochemical leak of Donggang Petrochemical Chemical Industry Co., Ltd. in the city of Quanzhou, Fujian Province as an example. During this leak, petrochemicals spread to beaches, causing great harm to the coastal environment, local industries, and residents. The disaster events of the oil spill were mapped to log transition activities to clearly describe the effects of the petrochemical spill on the environment and the lives of the nearby residents. Tables 20 and 21 show the execution traces of this oil spill disaster and the semantic description of transition activities, respectively.

Table 20 presents all the execution traces of the oil spill disaster event, denoted as $L =< \sigma_1, \sigma_2 \cdots \sigma_{29} >$. The study of the dynamic evolution of the disaster chain from the perspective of the event logs intuitively obtains the primary disaster, secondary disaster, and derived disaster. Disaster event t_0, as the primary disaster of the overall disaster chain, triggers a series of subsequent disaster events. Taking event trace B as an example, the dynamic transformational relations of the disaster chain are shown in Figure 14. The chain is a typical causal disaster chain, which is the most common type of disaster chain in the logs.

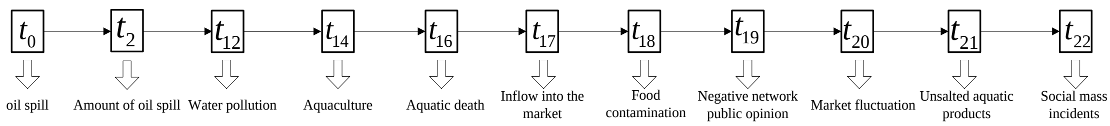

Figure 14. Dynamic conversion diagram of event trace σ_4.

Table 20. Disaster event logs of the oil spill.

Event Logs L_{dis}			
σ_1	$t_0t_1t_9t_{10}t_{11}$	σ_{16}	$t_0t_6t_{38}t_{39}$
σ_2	$t_0t_1t_{10}t_9t_{11}$	σ_{17}	$t_0t_7t_{40}$
σ_3	$t_0t_2t_{12}t_{13}t_{15}$	σ_{18}	$t_0t_8t_{46}t_{41}t_{42}t_{43}t_{44}t_{45}$
σ_4	$t_0t_2t_{12}t_{14}t_{16}t_{17}t_{18}t_{19}t_{20}t_{21}t_{22}$	σ_{19}	$t_0t_8t_{46}t_{41}t_{42}t_{43}t_{45}t_{44}$
σ_5	$t_0t_2t_{12}t_{14}t_{16}t_{17}t_{18}t_{19}t_{20}t_{22}t_{21}$	σ_{20}	$t_0t_8t_{46}t_{41}t_{42}t_{44}t_{43}t_{45}$
σ_6	$t_0t_2t_{12}t_{14}t_{16}t_{17}t_{18}t_{19}t_{21}t_{20}t_{22}$	σ_{21}	$t_0t_8t_{46}t_{41}t_{42}t_{44}t_{45}t_{43}$
σ_7	$t_0t_2t_{12}t_{14}t_{16}t_{17}t_{18}t_{19}t_{21}t_{22}t_{20}$	σ_{22}	$t_0t_8t_{46}t_{41}t_{42}t_{45}t_{43}t_{44}$
σ_8	$t_0t_2t_{12}t_{14}t_{16}t_{17}t_{18}t_{19}t_{22}t_{20}t_{21}$	σ_{23}	$t_0t_8t_{46}t_{41}t_{42}t_{45}t_{44}t_{43}$
σ_9	$t_0t_2t_{12}t_{14}t_{16}t_{17}t_{18}t_{19}t_{22}t_{21}t_{20}$	σ_{24}	$t_0t_8t_{41}t_{46}t_{42}t_{43}t_{44}t_{45}$
σ_{10}	$t_0t_3t_{23}t_{24}t_{26}$	σ_{25}	$t_0t_8t_{41}t_{46}t_{42}t_{43}t_{45}t_{44}$
σ_{11}	$t_0t_3t_{23}t_{25}t_{27}t_{28}$	σ_{26}	$t_0t_8t_{41}t_{46}t_{42}t_{44}t_{43}t_{45}$
σ_{12}	$t_0t_4t_{29}t_{30}$	σ_{27}	$t_0t_8t_{41}t_{46}t_{42}t_{44}t_{45}t_{43}$
σ_{13}	$t_0t_4t_{29}t_{31}t_{32}t_{33}$	σ_{28}	$t_0t_8t_{41}t_{46}t_{42}t_{45}t_{43}t_{44}$
σ_{14}	$t_0t_5t_{34}t_{35}t_{36}t_{37}t_{38}t_{39}$	σ_{29}	$t_0t_8t_{41}t_{46}t_{42}t_{45}t_{44}t_{43}$
σ_{15}	$t_0t_5t_{34}t_{35}t_{37}t_{36}t_{38}t_{39}$		

Table 21. Semantic descriptions.

Symbol	Semantic	Symbol	Semantic	Symbol	Semantic
t_0	Oil spill	t_{16}	Aquatic death	t_{32}	Injury and death
t_1	Nature reserve	t_{17}	Inflow into the market	t_{33}	Negative network public opinion
t_2	Amount of oil spill	t_{18}	Food contamination	t_{34}	Maritime traffic disruption
t_3	Volatility of oil	t_{19}	Negative network public opinion	t_{35}	Unconditional trigger
t_4	Flammability of oil	t_{20}	Market fluctuation	t_{36}	Freight break
t_5	Maritime traffic area	t_{21}	Unsalted aquatic products	t_{37}	Tourism damage
t_6	Scenic area	t_{22}	Social mass incidents	t_{38}	Unconditional trigger
t_7	Controversial area	t_{23}	Air pollution	t_{39}	Service industry damage
t_8	Shoreline	t_{24}	Oil spill responders	t_{40}	Foreign-related events
t_9	Ecological destruction	t_{25}	Coastal residential area	t_{41}	Shore beach pollution
t_{10}	Wildlife death	t_{26}	responder poisoning	t_{42}	Coastal industrial area
t_{11}	Animal epidemic event	t_{27}	Mass poisoning event	t_{43}	Industrial production damage
t_{12}	Water pollution	t_{28}	Negative network public opinion	t_{44}	School suspension
t_{13}	Urban water supply area	t_{29}	Hazardous chemicals explosion	t_{45}	Transport industry damage
t_{14}	Aquaculture	t_{30}	Negative network public opinion	t_{46}	Soil pollution
t_{15}	Urban water supply interruption	t_{31}	Human existence		

There are many more event traces and transition activities in the disaster event logs, corresponding to a huge amount of data and a complex model. In view of the detailed introduction and calculation of the behavioral profile relations and log minimum successor relations described in Section 3, this section does not give the complete behavioral profile matrix but only calculates and analyzes some of the event traces. Taking the sequence of traces σ_4 to σ_9 as an example, we calculate the mutual behavioral profile relations and the log minimum successor relations of the transitions $t_{20}t_{21}t_{22}$, which satisfy $BP_{(t_{20},t_{21})} = \|$, $MSR_{(t_{20},t_{21})} = 1$, $BP_{(t_{20},t_{22})} = \|$, $MSR_{(t_{20},t_{22})} = 1$, $BP_{(t_{21},t_{22})} = \|$, and $MSR_{(t_{21},t_{22})} = 1$. To gain a more intuitive understanding, the behavioral relations of $t_{20}t_{21}t_{22}$ are simulated using the Petri net model. The model structure shown in Figure 15 is obtained.

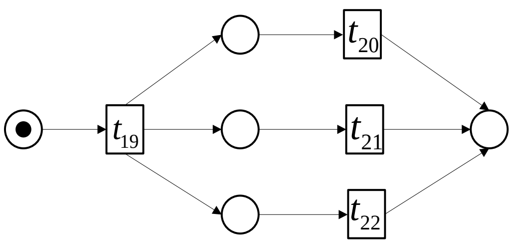

Figure 15. Petri net model of a concurrent disaster chain.

Similarly, the calculation rules and determination method in Algorithm 1 can be used to calculate exclusive and coupled disaster chains. However, it should be noted that for coupled disaster chains, the cause is the coupling of multiple disasters, and their risks and damage degree are thus higher [27]. This paper mainly introduces the method of change mining using log behavioral profile relations and does not discuss how to conduct risk assessment research for a coupled disaster chain or how to transform strong coupling into weak coupling or zero coupling, which are topics to be explored in further research.

The logical order of disaster events is usually different in different scenarios. Affected by different environmental and social factors, the previous and subsequent events of a disaster chain are slightly different: $\sigma_{10} = t_0t_3t_{23}t_{24}t_{26}$ and $BP(t26, t28) = \varnothing$ in L_{dis}. The trace indicates that the oil spill caused responder poisoning during the rescue process. In different situations, responder poisoning can directly lead to the occurrence of t_{28} (negative public opinions on the network). Therefore, $\sigma_{10'} = t_0t_3t_{23}t_{24}t_{26}t_{28}$ includes behavioral relations, denoted $BP(t26, t28) = \rightarrow, MSR(t26, t28) = 1 \wedge r(t26, t28) > 0$. An analysis of Algorithm 1 reveals that the occurrence of the switch hidden transition leads to behavioral changes. We can clearly observe that an oil spill eventually leads to the occurrence of social mass incidents from σ_4 to σ_9. The social mass incidents then lead to the continued fermentation of negative network public opinion. On the basis of this dynamic description, an execution sequence of $L =< \sigma_4, \sigma_5, \sigma_6, \sigma_7, \sigma_8, \sigma_9 >$ occurs, where $\sigma_4 = t_0t_2t_{12}t_{14}t_{16}t_{17}t_{18}\dot{t}_{19}\dot{t}_{20}\dot{t}_{21}\dot{t}_{22}$, $\sigma_5 = t_0t_2t_{12}t_{14}t_{16}t_{17}t_{18}\dot{t}_{19}\dot{t}_{20}\dot{t}_{22}\dot{t}_{21}$, $\sigma_6 = t_0t_2t_{12}t_{14}t_{16}t_{17}t_{18}\dot{t}_{19}\dot{t}_{21}\dot{t}_{20}\dot{t}_{22}$, $\sigma_7 = t_0t_2t_{12}t_{14}t_{16}t_{17}t_{18}\dot{t}_{19}\dot{t}_{21}\dot{t}_{22}\dot{t}_{20}$, $\sigma_8 = t_0t_2t_{12}t_{14}t_{16}t_{17}t_{18}\dot{t}_{19}\dot{t}_{22}\dot{t}_{20}\dot{t}_{21}$, and $\sigma_9 = t_0t_2t_{12}t_{14}t_{16}t_{17}t_{18}\dot{t}_{19}\dot{t}_{22}\dot{t}_{21}\dot{t}_{20}$. We deduce from Algorithm 1 that the occurrence of the loop hidden transition may lead to changes in the disaster logs. However, because the disaster chain is a concurrent disaster chain, the location of the hidden transition has three possibilities, which are marked with dotted boxes in Figures 16 and 17 showing the specific simulation results.

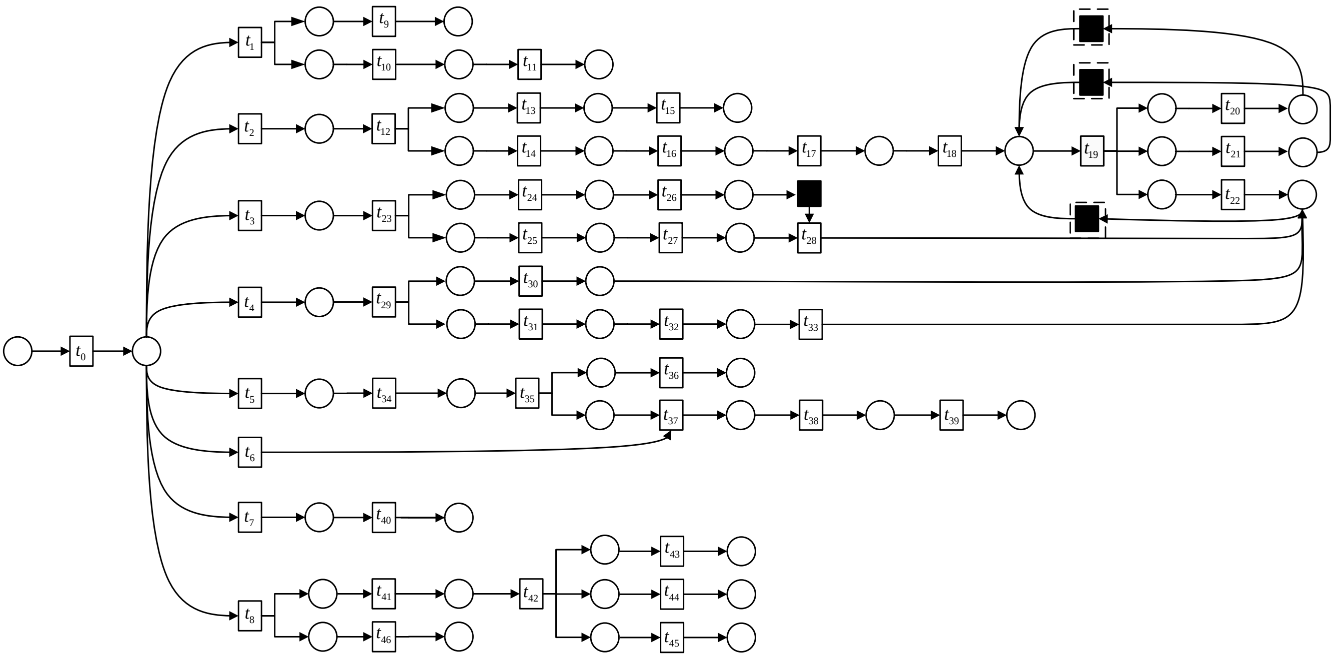

Figure 16. Petri net model with hidden transitions for oil spill disaster events.

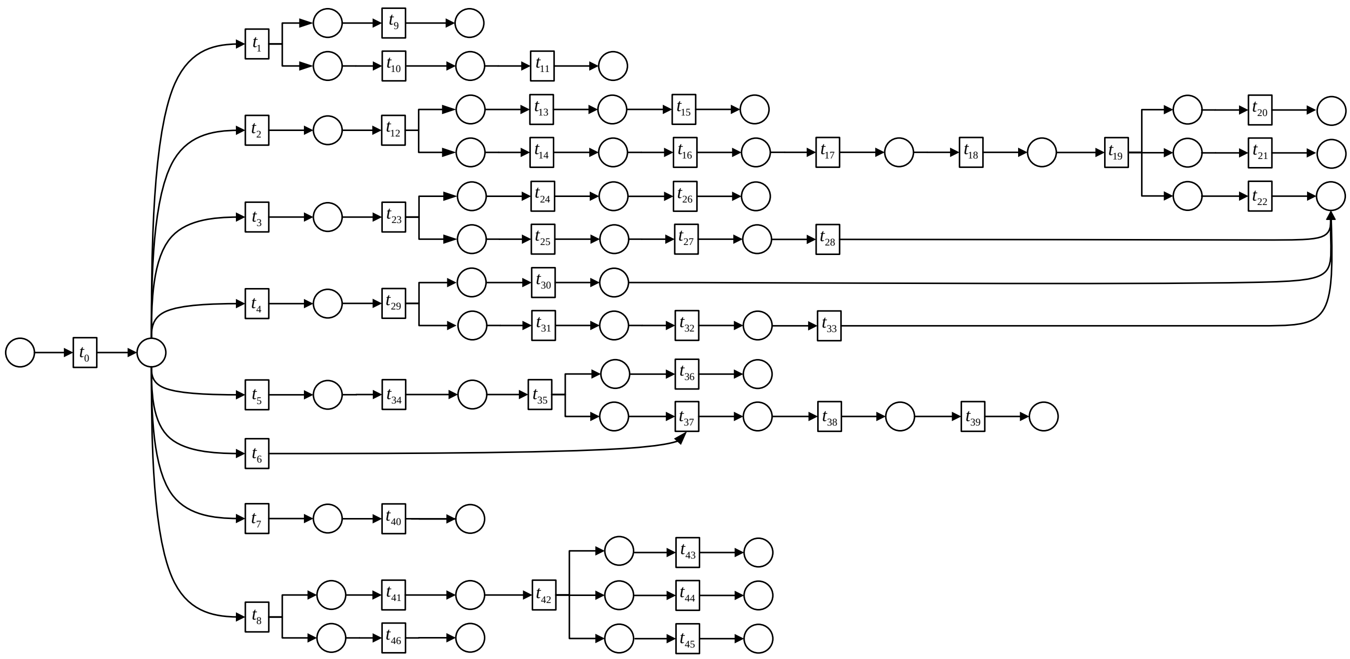

Figure 17. Original Petri net model for oil spill disaster events.

Figures 16 and 17 are Petri net models of the disaster events with and without hidden transitions. Using Algorithm 1, we can mine the dynamic evolution process of disaster chains in different scenarios when only the disaster event logs are given, and we can clarify the basic types of disaster chain. For example, if there is a higher risk of a coupled disaster chain, the decision maker can make a reasonable judgment to cut this chain for disaster mitigation.

7. Conclusions

This paper proposed a method for mining behavioral changes based on complete logs with hidden transitions. Firstly, to capture the dependencies between transition activities more accurately, the method of calculating the log minimum successor relations was defined. Secondly, we proposed classification rules of hidden transitions and calculation rules of log-weighted coefficients. Process models with strict, and-split, and-join, skip, loop, and switch hidden transitions were then identified by combining the log behavioral profile, log minimum successor relation, log-weighted coefficient and changed behavioral relations. Finally, the correctness and feasibility of the approach were verified in experiments and a case study. There are three main aspects of future research: (1) improving the methodology so that it can be applied to non-freely chosen structures; (2) considering how to correctly mine hidden transitions and changed behavioral relations when the behavioral relations of logs are incomplete, and (3) considering further application of the methodology to disaster chain risk assessment by adding probabilistic constraints to estimate the probability of subsequent disaster events occurring.

Author Contributions: S.S.: conceptualization, writing—original draft, data curation. Q.L.: validation, writing—review and editing. All authors have read and agreed to the published version of the manuscript.

Funding: The research was supported by the National Natural Science Foundation of China (grant numbers 71971121), NUIST-UoR International Research Institute, the Major Project Plan of Philosophy and Social Sciences Research at Jiangsu University (grant number 2018SJZDA038). Additionally, we thank Liwen Bianji (Edanz) (www.liwenbianji.cn accessed on 28 December 2022) for editing the language of a draft of this manuscript.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The data presented in this study are available on request from the corresponding author.

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

References

1. Choueiri, A.C.; Santos, E.A.P. Discovery of path-attribute dependency in manufacturing environments: A process mining approach. *J. Manuf. Syst.* **2021**, *61*, 54–65. [CrossRef]
2. Ko, J.; Comuzzi, M. Detecting anomalies in business process event logs using statistical leverage. *Inf. Sci.* **2021**, *549*, 53–67. [CrossRef]
3. Fang, X.W.; Yan, Y.; Liu, X. An Analysis Method about Change Region of Business Process Model Based On Action Pattern. *Int. J. Multimed. Eng.* **2014**, *5*, 251–262. [CrossRef]
4. Weidmann, M.; Alvi, M.; Koetter, F. Business process change management based on process model synchronization of multiple abstraction levels. In Proceedings of the IEEE International Conference on Service-Oriented Computing and Applications, Irvine, CA, USA, 12–14 December 2011.
5. Weidlich, M.; Mendling, J.; Weske, M. Propagating change between aligned process models. *J. Syst. Software* **2012**, *85*, 1885–1898. [CrossRef]
6. Fang, X.W.; Liu, L.; Liu, X.W. Analyzing method of change region in BPM based on modules of Petri net. *ITJ* **2013**, *12*, 1655–1659. [CrossRef]
7. Dam, H.K.; Michael, W. An Agent-Oriented Approach to Change Propagation in Software Maintenance. *Auton. Agent. Multi-Ag.* **2011**, *23*, 384–452. [CrossRef]
8. Assy, N.; Gaaloul, W.; Defude, B. Mining Confgurable Process Fragments for Business Process Design. In Proceedings of the 9th International Conference on Design Science Research in Information Systems and Technology, Miami, FL, USA, 22 May 2014.
9. Gao, X.; Chen, Y.; Ding, Z.Z. Process Model Fragmentization, Clustering and Merging: An Empirical Study. In Proceedings of the 11th International Conference on Business Process Management, Beijing, China, 26–30 August 2013.
10. Pourmasoumi, A.; Kahani, M.; Bagheri, E. Mining variable fragments from process event logs. *Inform. Syst. Front.* **2017**, *19*, 1423–1443. [CrossRef]
11. Fang, H.; Sun, S.Y.; Fang, X.W. Behavior change mining methods based on incomplete logs conjoint occurrence relation. *CIMS* **2020**, *26*, 1887–1895. (In Chinese)
12. Nolle, T.; Luettgen, S.; Seeliger, A. BINet: Multi-perspective business process anomaly classifcation. *Inform. Syst.* **2022**, *103*, 101458. [CrossRef]
13. 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.
14. Nolle, T.; Luettgen, S.; Seeliger, A. Analyzing business process anomalies using autoencoders. *J. Mach. Learn.* **2018**, *107*, 1875–1893. [CrossRef]
15. Weinzierl, S.; Wolf, V.; Pauli, T. Detecting temporal workarounds in business processes–A deep-learning-based method for analysing event log data. *J. J. Bus. Anal.* **2022**, *5*, 76–100. [CrossRef]
16. Harl, M.; Weinzierl, S.; Stierle, M. Explainable predictive business process monitoring using gated graph neural networks. *J. Decis. Syst.* **2020**, *29*, 312–327. [CrossRef]
17. Liu, B.; Hsu, W.; Han, H.S. Mining changes for real-life applications. In Proceedings of the International Conference on Data Warehousing and Knowledge Discovery, London, UK, 4–6 September 2000.
18. Luna, R.; Balakrishnan, N.; Dagli, C.H. Postearthquake recovery of a water distribution system: Discrete event simulation using colored petri nets. *J. Infrastruct. Syst.* **2011**, *17*, 25–34. [CrossRef]
19. Zhang, Y.; Huang, S.H.; Dang, D.; Ruan, H. Evaluation of Natural Disaster Emergency Response Procedure Based on Petri Net. *Appl. Mech. Mater* **2013**, *339*, 236–241. [CrossRef]
20. Qi, C.; Luo, L.J. Analysis of Flood Disaster Case Based on Event Chain and Generalized Stochastic Petri Nets. *J. Wuhan. Univ. Technol.* **2017**, *39*, 130–134.
21. Zhao, Q.; Wang, J. Disaster Chain Scenarios Evolutionary Analysis and Simulation Based on Fuzzy Petri Net: A Case Study on Marine Oil Spill Disaster. *IEEE Access.* **2019**, *99*, 183010–183023. [CrossRef]
22. Ko, J.; Comuzz, M. Keeping our rivers clean: Information-theoretic online anomaly detection for streaming business process events. *Inform. Syst.* **2021**, *104*, 101894. [CrossRef]
23. Wu, Z.H. *Introduction to Petri*; Machinery Industry Press: Beijing, China, 2006; pp. 1–3. (In Chinese)
24. Van der Aalst, W. Process Discovery: Capturing the Invisible. *IEEE. Comput. Intell. M.* **2010**, *5*, 28–41. [CrossRef]
25. Weidlich, M.; Polyvyanyy, A.; Desai, N. Process Compliance Measurement Based on Behavioural Profles. *Inform. Syst.* **2011**, *36*, 1009–1025. [CrossRef]

26. Kunze, M.; Weidlich, M.; Weske, M. Querying process models by behavior inclusion. *Softw. Syst. Model.* **2015**, *14*, 1105–1125. [CrossRef]
27. Lin, Z.; Zhang, B.; Guo, J. Analysis of a Water-Inrush Disaster Caused by Coal Seam Subsidence Karst Collapse Column under the Action of Multi-Field Coupling in Taoyuan Coal Mine. *Cmes-Comp. Model. Eng.* **2021**, *126*, 311–330.

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.