IEICE TRANS. FUNDAMENTALS, VOL.E98–A, NO.2 FEBRUARY 2015

PAPER Special Section on Mathematical Systems Science and its Applications

Diagnosis of Stochastic Discrete Event Systems Based on *N*-gram Models

Miwa YOSHIMOTO[†], *Nonmember*, **Koichi KOBAYASHI**[†], *and* **Kunihiko HIRAISHI**[†a)], *Members*

SUMMARY In this paper, we present a new method for diagnosis of stochastic discrete event system. The method is based on anomaly detection for sequences. We call the method *sequence profiling* (SP). SP does not require any system models and any system-specific knowledge. The only information necessary for SP is event logs from the target system. Using event logs from the system in the normal situation, *N*-gram models are learned, where the *N*-gram model is used as approximation of the system behavior. Based on the *N*-gram model, the diagnoser estimates what kind of faults has occurred in the system, or may conclude that no faults occurs. Effectiveness of the proposed method is demonstrated by application to diagnosis of a multi-processor system.

key words: *stochastic discrete event systems, diagnosis, N-gram models*

1. Introduction

The problem of failure diagnosis has been studied from 1970's and there are various schemes for it. For systems with continuous state variables, fault detection based on analytical modeling was proposed [1], [2]. In this approach, faults are detected by comparing actual measurements with their predicted values. Rule-based diagnosis (RBD) is a method for detecting and specifying system faults based on empirical rules obtained from accumulation of expert knowledge [3]. While computational cost of RBD is low for simple systems, it is hard to build complete rule bases for large and complex systems. Moreover, RBD is not suitable for autonomous fault detection.

In contrast, model-based diagnosis (MBD) is a method using system models that describe causal relation on event occurrence. There are many literature on MBD for discrete event systems (DES) (e.g., [4]). Diagnosis for stochastic DES and decentralized DES is also studied [5]–[8]. In MBD, faults are detected by comparing event logs observed in the actual system with those provided by the system model. This process is performed on finite-state automata called *diagnosers*. One of advantages of MBD is that it does not require any empirical rules depending on the target system. Moreover, MBD can be applied at low computational cost once the diagnoser is constructed. This is a desirable feature for on-line autonomous failure detection. However, MBD has several weaknesses described as follows.

The first weakness is that the size of the diagnoser can be doubly exponential in the size of the system description in the worst case. This can make the diagnoser practically impossible to generate. To overcome the problem on computational cost, SAT-based diagnosis was proposed [9]. Diagnosis based on behavior reconstruction, which is an intensional representation of all behaviors that are consistent with the system observation, was also proposed [10].

The second weakness is that MBD requires an accurate and detailed model of the target system. However, when the configuration of the target system changes dynamically, such as systems operated on virtual environments like in cloud systems, it is difficult to obtain accurate models for the entire system. Updating the system model is hard to be done in online environment.

In this paper, we present a new method for the diagnosis of stochastic discrete event system. We call the method *sequence profiling* (SP). The method is based on *anomaly detection technique* [11]. SP does not require any system models and any system-specific knowledge. The only information necessary for SP is event logs from the target system. In this sense, the proposed approach is not classified as MBD.

The overview of the proposed approach is illustrated in Fig. 1. Using event logs from the system in the normal situation, i.e., the situation in which no fault occurs, a probabilistic model that represents the system behavior is constructed (Learning Phase). Based on the learned behavior, the diagnoser estimates what kind of faults has occurred in the system, or may conclude that no faults occurs (Diagnosis Phase). In the SP, we assume that *faults in the system appear only in the probability of event occurrence*.

At the learning phase, an *N*-gram model that approximates the behavior of the target system is computed. The *N*-gram model was originally introduced by Shannon [12], and are mainly used for natural language processing [13]. Recently it is applied to text mining and web search. In the proposed method, *N*-gram models are used for discovering

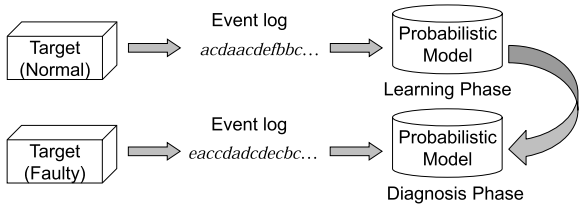

Fig. 1 Overview of the proposed approach.

Manuscript received April 8, 2014.
Manuscript revised June 27, 2014.

[†]The authors are with the School of Information Science, Japan Advanced Institute of Science and Technology, Nomi-shi, 923-1292 Japan.
a) E-mail: hira@jaist.ac.jp

DOI: 10.1587/transfun.E98.A.618

discrepancy between the observed event logs and the behavior of the system in the normal situation. The discrepancy is measured by *the specificity* of short event sequences in the event logs, where the short sequences correspond to local activities of the system.

As a similar approach based on learning, there exists a method using probabilistic models representing relationship between event sequences and potential faults [14]. In this method, it is assumed that faulty behavior appears only in the interevent time, and probability distribution function of the interevent time is estimated for every combination of faults. In the proposed method, however, it is assumed that faulty behavior appears in event sequences, i.e., the order of event occurrences. Process mining [15] is a technique that is used for estimating system models from event logs. The aim of process mining is to obtain correct system models, while in the proposed approach models approximating system behavior are computed and used for identifying faults. In this sense, the models in the proposed method becomes relatively simpler than the correct model.

This paper is organized as follows. In Sect. 2, some notations and definitions on formal languages and automata are given. In Sect. 3, approximation of probabilistic automata by *N*-gram models is explained. In Sect. 4, the diagnosis problem studied in this paper is formally stated. The proposed method is presented in Sect. 5. Comparison with existing methods for anomaly detection is also described. In Sect. 6, effectiveness of the proposed approach is demonstrated by application to diagnosis of a multi-processor system. Section 7 is the conclusion.

2. Preliminaries

Let Σ be a finite set of symbols and let Σ^* denote the set of all finite sequences over Σ. Let s be a sequence over Σ. We write $|s|$ to denote the length of s. The i-th symbol of s is denoted by $s_{[i]}$ and the subsequence of s from the i-th position to the j-th position is denoted by $s_{[i,j]}$. For a positive integer k, let $\Sigma^k = \{s \in \Sigma^* \mid |s| = k\}$ denote the set of all sequences of length k.

Let s and w be sequences over Σ such that $|s| < |w|$. The number of occurrences of s as a subsequence of w is denoted by $O_s(w)$, i.e.,

$$O_s(w) := |\{(i, j) \mid 1 \le i \le j \le |w|, w_{[i,j]} = s\}|. \quad (1)$$

We assume that the target system is modeled by a (nondeterministic) probabilistic automaton $G = (X, \Sigma \cup \{\varepsilon\}, \delta, P, x_0, F)$, where

- $X = \{x_1, \cdots, x_n\}$ is the set of states,
- Σ is the set of symbols and ε is the empty sequence,
- $\delta \subseteq X \times \Sigma \times X$ is the transition relation,
- $P : X \times \Sigma \times X \to [0, 1]$ is the function defining probability of state transition, i.e., $P(x_i, \sigma, x_j)$ is the probability that event σ occurs in state x_i and the state changes to x_j after the occurrence. We require that $P(x_i, \sigma, x_j) = 0$ if $(x_i, \sigma, x_j) \notin \delta$, and $\sum_{\sigma \in \Sigma, x_j \in X} P(x_i, \sigma, x_j) = 1$ for each state x_i,
- $x_0 \in X$ is the initial state,
- $F \subseteq X$ is the set of final states.

We will simply write P^σ_{ij} to denote $P(x_i, \sigma, x_j)$. Moreover, let $P^\sigma_{i\bullet} := \sum_j P^\sigma_{ij}$. As usual, state transition relation is extended to $\delta \subseteq X \times \Sigma^* \times X$ by (i) $(x_i, \varepsilon, x_i) \in \delta$ and (ii) $(x_i, s\sigma, x_k) \in \delta$ if $(x_i, s, x_j) \in \delta$ and $(x_j, \sigma, x_k) \in \delta$.

The underlying Markov chain of G is given by the set of states X, and the probability for each pair of states $P_{ij} = \sum_{\sigma \in \Sigma} P^\sigma_{ij}$.

3. Approximating Probabilistic Automata by N-gram Models

Given a sequence w over Σ, the maximum likelihood estimation of the N-gram model for w is given as the following probabilities for all $y \in \Sigma^{N-1}$ and all $\sigma \in \Sigma$:

$$Pr(\sigma|y) := \frac{O_{y\sigma}(w)}{\sum_{\sigma' \in \Sigma} O_{y\sigma'}(w)} \quad (2)$$

Given a sequence w, N-grams are extracted and the number of occurrences of each N-gram is counted. This procedure requires $O(|w|)$ time. Next the conditional probability for each y and σ is computed by (2). The number of conditional probabilities to be computed is at most $|\Sigma|^N$. Summing up $O_{y\sigma'}(w)$'s in the denominator requires $O(|\Sigma|)$ time for each $y \in \Sigma^{N-1}$. Hence the total time for computing an N-gram model is $O(|w| + |\Sigma|^N)$.

Let $G = (X, \Sigma \cup \{\varepsilon\}, \delta, P, x_0, F)$ be a probabilistic automaton. Suppose that a finite sequence s has occurred from some *unknown state* of G. We consider the problem of estimating the next event to occur. Let X_y denote the set of such states that sequence y has occurred just before reaching it, i.e.,

$$X_y = \{x_j \mid \exists x_i \in X : (x_i, y, x_j) \in \delta\} \quad (3)$$

Next we define an equivalence $\sim$ on X by $x_i \sim x_j \Leftrightarrow \forall \sigma \in \Sigma : P^\sigma_{i\bullet} = P^\sigma_{j\bullet}$. We say that G is *predictable* for sequence y if all states in X_y are equivalent w.r.t. $\sim$. Note that if $|X_y| = 1$ then the automaton is obviously predictable for y. Moreover, G is called *k-predictable* if G is predictable for all sequence y of length k.

Whether G is k-predictable or not is determined by its structure. The automaton in Fig. 2 is 1-predictable. However, the automaton in Fig. 3 is not k-predictable for any k because G is not predictable for sequences b^*a of arbitrary length ($X_{b^*a} = \{x_1, x_2\}$).

Given a probabilistic automaton G, we say that an N-gram model *approximates* G if for any sequence $y \in \Sigma^{N-1}$ that has occurred in G, the probability that an event σ will occur after y is $Pr(\sigma|y)$. We show the cases in which such an N-gram exists.

Suppose that a probabilistic automaton G is $(N-1)$-predictable. Then we can obtain the following N-gram model that correctly gives the probability: for each $y \in \Sigma^{N-1}$ and $\sigma \in \Sigma$,

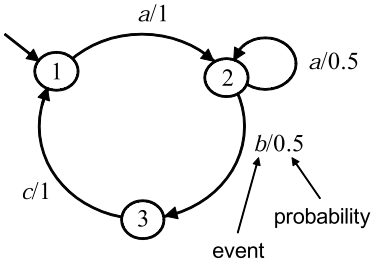

Fig. 2 Probabilistic automaton: example 1.

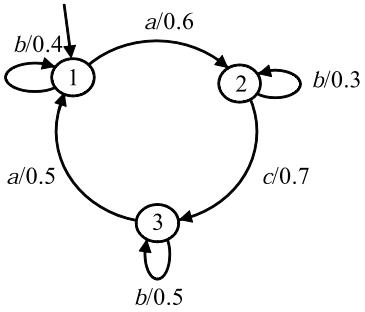

Fig. 3 Probabilistic automaton: example 2.

$$Pr(\sigma|y) := P^{\sigma}_{i\bullet} \tag{4}$$

where $x_i \in X_y$.

Suppose that G is not $(N-1)$-predictable and $|X_y| > 1$ for some $y \in \Sigma^{N-1}$. If the automaton has the steady state and the stationary distribution $\pi = (\pi_1, \cdots, \pi_n)$ is known, where π_i is the probability that the system is in state x_i, then we can obtain the following N-gram model: for each $y \in \Sigma^{N-1}$ and $\sigma \in \Sigma$,

$$Pr(\sigma|y) = \sum_{x_i \in X_y} \left(\pi_i / \sum_{x_j \in X_y} \pi_j \right) \cdot P^{\sigma}_{i\bullet} \tag{5}$$

Figure 3 is a probabilistic automaton whose underlying Markov chain is ergodic. This automaton has the unique stationary distribution $\pi = (35/107, 30/107, 42/107)$ as the solution of equations $\pi = \pi\mathbf{P}$, $\sum_i \pi_i = 1$, where $\mathbf{P} = [P_{ij}]$ is the transition probability matrix. After an occurrence of ab, possible states are 1 and 2. Therefore, the conditional probability $Pr(b|ab)$ is computed by

$$Pr(b|ab) = \frac{\pi_1}{\pi_1 + \pi_2} \cdot 0.4 + \frac{\pi_2}{\pi_1 + \pi_2} \cdot 0.3 = 23/65.$$

The following is a direct consequence of the theory of Markov chains.

Theorem 3.1: Let G be a probabilistic automaton whose underlying Markov chain is ergodic. Then the probabilities by (2) approaches to the probabilities by (5) as the length of w increases.

4. Fault Identification Problem

Let $G_i, i = 0, 1, \cdots, m$ be probabilistic automata, where G_0 is an automaton that represents the system in the normal situation, and let $G_i, i = 1, \cdots, m$ be automata for faulty situations. We assume that G_i's are the same automaton except that *probabilities of some state transitions are different*. Such faults correspond to malfunction of some part of the system. A typical example of such faults will be shown in Sect. 6.

In this paper, fault detection of stochastic discrete event systems is considered as the following problem:

Definition 4.1 (Fault Identification Problem): Given a sequence w_{test} of events over Σ, choose a system model from $G_i, i = 0, 1, \cdots, m$ that most likely generates w_{test}.

This problem is exactly the same as the anomaly detection for sequences [16], except that we put the assumption on type of faults.

If the normal/faulty system models $G_i, i = 0, 1, \cdots, m$, are given, then the problem is solvable by the model-based diagnosis. In this paper, we put the following assumptions on the input data:

- System models $G_i, i = 0, 1, \cdots, m$, are unknown or hard to obtain or too large to handle.
- The System model in the normal situation has the steady state.
- Event logs from the models $G_i, i = 0, 1, \cdots, m$, are available.
- Event logs may not be those from the initial state. In other words, event logs are not synchronized.

These assumptions are valid in the situation that event logs are automatically recorded in the system and they are connected to specific faulty cases when some faults happen. Such a situation can be seen in enterprise information systems and cloud systems.

We can also consider the problem under partial observation. Let $G = (X, \Sigma \cup \{\varepsilon\}, \delta, P, x_0, F)$ be a probabilistic automaton. *An observation map* is a function $O : \Sigma \to \Pi \cup \{\varepsilon\}$, where Π is a different set of symbols. O is extended to $O : \Sigma^* \to (\Pi \cup \{\varepsilon\})^*$ in the usual way. Let $O(G)$ denote the probabilistic automaton $G' = (X, \Pi \cup \{\varepsilon\}, \delta', P', x_0, F)$ obtained by

- $(x_i, \sigma', x_j) \in \delta'$ if $\exists \sigma \in \Sigma : (x_i, \sigma, x_j) \in \delta$ and $O(\sigma) = \sigma'$,
- $P'(x_i, \sigma', x_j) := \sum_{\sigma : O(\sigma) = \sigma'} P(x_i, \sigma, x_j)$.

Since $O(G)$ is a probabilistic automaton, we can consider the fault identification problem for the case that $O(w_{test})$ is observed instead of w_{test}.

Let $G_i = (X_i, \Sigma \cup \{\varepsilon\}, \delta_i, P_i, x_0, F_i), i = 1, 2$ be a set of probabilistic automaton defined over the same set of events. We say that G_1 and G_2 are *indistinguishable* w.r.t N-gram model if G_1 and G_2 give the same N-gram model based on (5). If $O(G_i), i = 0, 1, \cdots, m$ are indistinguishable w.r.t. the N-gram model, then no algorithm based on the N-gram model can answer the fault identification problem.

5. Sequence Profiling

We present the proposed diagnosis method, called sequence profiling.

5.1 Sequence Specificity

We first define *the specificity* of sequences based on the N-gram model. Computing specificities is the main part of the sequence profiling. We assume that the k-gram model is obtained for all $1 \leq k \leq N$. We will call the collection of these models the $N^{\leq}$-gram model. Based on this model, conditional probability $Pr(s|y)$ for y of any length is determined by

$$Pr(s|y) = Pr(s|y_{[j,\,|y|]}) \tag{6}$$

where $j = \max\{|y| - N + 2,\ 1\}$.

Suppose that we already have an $N^{\leq}$-gram model for G_0. Let s be a sequence of length r. Then the conditional probability that sequence s appears after sequence y is given by the following recursion:

$$Pr(s \mid y) := Pr(s_{[1]} \mid y) \cdot Pr(s_{[2,\,|s|]} \mid y s_{[1]}) \tag{7}$$

We show an example. Using a 4-gram model, $Pr(abx|xyz)$ is obtained by

$$\begin{aligned} &Pr(abc|xyz) \\ &= Pr(a|xyz) \cdot Pr(bc|xyza) \\ &= Pr(a|xyz) \cdot Pr(b|xyza) \cdot Pr(c|xyzab) \\ &= Pr(a|xyz) \cdot Pr(b|yza) \cdot Pr(c|zab) \end{aligned}$$

where $Pr(b|xyza) = Pr(b|yza)$ and $Pr(c|xyzab) = Pr(c|zab)$ as we have assumed in (6).

The expected number of times s occurs in w is computed by

$$E_s(w) = \sum_{y \in \Sigma^{N-1}} O_y(w) \cdot Pr(s|y) \tag{8}$$

Now we define the specificity of sequence s in w w.r.t. the $N^{\leq}$-gram model as follows:

$$d_s^N(w) = \log \frac{O_s(w)}{E_s(w)} \tag{9}$$

If $d_s^N(w)$ is positive (negative), then s appears more (less) often than the average.

Given a nonnegative integer r, let $D^{r,N}(w)$ denote the $|\Sigma^r|$-dimensional vector each component of which is $d_s^N(w)$ for $s \in \Sigma^r$. We can expect $D^{r,N}(w)$ captures some characteristics of sequence w, and we call it *the profile* of w w.r.t. the two parameters r and N. Note that the dimension of the specificity vector $D^{r,N}(w)$ is $|\Sigma^r| = |\Sigma|^r$. This is an exponential function of r, but is independent of the length $|w|$. Moreover, some of sequences in Σ^r may not occur in the target system, and therefore the actual dimension of $D^{r,N}(w)$ is usually smaller than $|\Sigma|^r$.

5.2 Sequence Profiling Algorithm

The detailed procedure of sequence profiling is described as follows:

1. Let w_{ref} be an event sequence generated by the system in the normal situation. Compute the $N^{\leq}$-gram model of w_{ref}.
2. Let w_i be an event sequence generated by model G_i, $i = 0, 1, \cdots, m$, where $|w_0|, |w_1|, \cdots, |w_m| \ll |w_{ref}|$. Based on the $N^{\leq}$-gram model, compute the specificity vector $D^{r,N}(w_i)$ for each w_i, $i = 0, 1, \cdots, m$.
3. Find the vector $D^{r,N}(w_i)$ that shows the highest correlation with $D^{r,N}(w_{test})$, and output the model G_i. This can be done by clustering of the specificity vectors, or by computing corelation coefficients between $D^{r,N}(w_{test})$ and $D^{r,N}(w_i)$, $i = 0, 1, \cdots, m$.

Different from researches on learning probabilistic models such as [17], obtaining accurate system models is not the objective of SP. The model learned from w_{ref} should be simpler but enough for enabling the fault identification.

5.3 Comparison With Existing Anomaly Detection Techniques

There are many results on anomaly detection [11]. Anomaly detection problems are classified by various features such as nature of data and anomaly types. The fault identification problem considered in this paper is classified as anomaly detection for sequences, where sequences are time series of symbolic data. For this class of problems, various existing techniques based on probabilistic modeling are compared with each other in [16]. The techniques used in the evaluation are (i) finite state automata based techniques [18], (ii) probabilistic suffix trees [19], (iii) sparse Markovian techniques [20], and (iv) hidden Markov model based techniques [21].

Finite state automata techniques are based on conditional probability for preceding N-grams with a fixed N, whereas probabilistic suffix trees can deal with conditional probability for variable-length sequences. Sparse Markovian techniques are more flexible in the sense that conditional probabilities are estimated based on a subset of symbols within the preceding N-grams. Hidden Markov models based techniques do not use conditional probabilities for preceding sequences. Given the number of hidden states, hidden state spaces are estimated for test sequences. The proposed method SP uses a technique similar to that used in the finite state automata techniques, i.e., SP uses conditional probabilities for preceding fixed-length N-grams.

Comparing with anomaly detection problems studied in literature, we put special assumptions on the target systems and also on type of faults, as described Sect. 4. Both the system in the normal situation and the system in faulty situation are represented by the same probabilistic automaton except that probabilities of some state transitions are

different. This means that whether some sequences appear in the test sequence or not does not contributes to detecting faults, and therefore we need to take the frequency of event occurrence into account. Moreover, if the change in the probabilities appear only in a limited number of state transitions, then the effect of the faults is also limited. As a result, the system behave normally in most of the time, but in some occasion the system shows unusual behavior.

In most of the existing techniques, anomaly scores are computed for the entire sequences, while SP focuses on fragments of event sequencers in order to detect localized anomaly in the test sequence. This is the main distinction of SP comparing with existing approaches. An anomaly score is computed for each sequence fragment and the entire sequence is characterized by the vector of the scores. We remark that the specificity is independent of the sequence length. This enable us to compare sequences having different lengths. Importance of such anomaly scores *normalized by the length* is discussed in [19].

6. Experimental Results

6.1 A Multi-Processor System

We apply the proposed method to a multi-processor system described in [9]. The system consists of 9 Processing Units (PUs) arranged on 3×3 grid layout. Figure 4 shows the state transition diagram of a single PU, and Fig. 5 shows the system configuration. When a fault occurs in some PU, the PU will reboot and simultaneously send a message (*reboot!*) to 4-neighbors. If a neighbor PU receives a reboot message (*reboot?*), then the PU will reboot (*IReboot*), too. After the rebooting process (*rebooting*), the PU returns to the normal operation (*IAmBack*). The number of states will be 6^9 =10,077,696.

We model the system by a stochastic Petri net (SPN) tool StpnPlay [22]. During the simulation, all event occurrences are recorded in a log. In faulty models, we give smaller values for stochastic exponential delays of event *reboot!*. Note that PUs may reboot even in the normal model, but the frequency of rebooting is lower than that in the faulty models.

Each event log is converted into a sequence of symbols, according to the following observation map:

- *done* of PU2, PU5, PU8 $\rightarrow A$
- *IReboot* of PU2 $\rightarrow R$
- *IReboot* of PU5 $\rightarrow S$
- *IReboot* of PU8 $\rightarrow T$
- *IAmBack* of PU2, PU5, PU8 $\rightarrow X$
- Other events: $\rightarrow \varepsilon$

We assume that faulty events are unobservable. Therefore, we need to detect the fault by observation of other events.

6.2 Test Data

We prepare one normal model (G_{nrm0}) and nine faulty models ($G_{abn1} - G_{abn9}$). Five event logs ($nrm01 - nrm05, abn11 - abn15, \cdots abn91 - abn95$) are obtained from each model. In addition, we prepare an event log $nrm0$ generated by the normal model, where $|nrm0| = 1,000,000$ and $|nrm0X| = |abn1X| = \cdots = |abn9X| = 100,000, X \in \{1, \cdots, 5\}$. These numbers are those before application of the observation map. In the experiments, it is observed that each length after application of the observation map is around 18% of the original length.

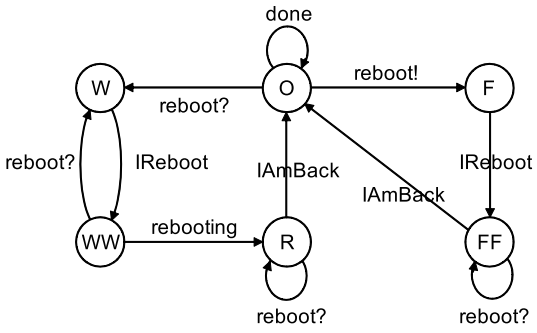

Fig. 4 One processing unit.

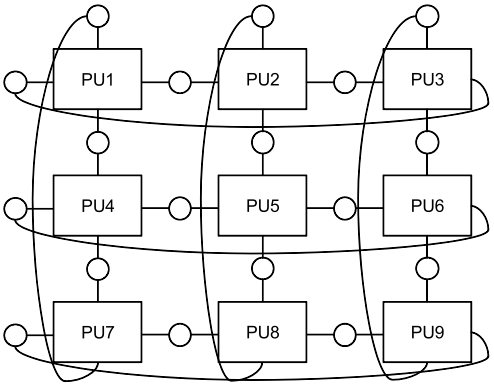

Fig. 5 System configuration.

From the event log $nrm0$, $N^{\leq}$-gram model is computed, where we use $N = 3$ in the experiments. The specificity is computed for short sequences of length $r = 3$, where we use (9) for the computation.

6.3 Preliminary Test

We first apply clustering analysis to classify the data into categories depending on similarities between the obtained vectors. The proposed approach is valid for fault identification if we succeed to partition the event logs into 10 groups ($nrm0X, abn1X, \cdots, abn9X$).

There are two approaches in the clustering analysis: one is the hierarchical clustering and the other is the non-hierarchical clustering. In the non-hierarchical clustering, the number of clusters is given beforehand, and the optimal clusters is computed under this constraint. We first show a result of non-hierarchical clustering based on k-means method. The number of clusters was set as 2 in order to divide the normal group ($nrm0X$) and the faulty groups. The

result is shown in Table 1. The result suggests that the normal group is not clearly separated from the faulty groups, but logs in the same group are classified into the same cluster.

6.4 Fault Identification by Clustering

In the hierarchical clustering analysis, the number of clusters is not fixed. The distance as degree of dissimilarity between clusters is computed. Using the distance, nearest clusters are unified into one cluster. This procedure is repeated until all clusters are unified into one cluster. The result of the unifying process is formulated as *a dendrogram*. We here use Euclidean distance as degree of dissimilarity. The result of the hierarchical clustering using average linkage concept is shown in Fig. 6.

Table 1 Result of non-hierarchical clustering.

Log ID	Cluster ID	Log ID	Cluster ID
*abn*11	2	*abn*61	1
*abn*12	2	*abn*62	1
*abn*13	2	*abn*63	1
*abn*14	2	*abn*64	1
*abn*15	2	*abn*65	1
*abn*21	1	*abn*71	1
*abn*22	1	*abn*72	1
*abn*23	1	*abn*73	1
*abn*24	1	*abn*74	1
*abn*25	1	*abn*75	1
*abn*31	2	*abn*81	2
*abn*32	2	*abn*82	2
*abn*33	2	*abn*83	2
*abn*34	2	*abn*84	2
*abn*35	2	*abn*85	2
*abn*41	1	*abn*91	1
*abn*42	1	*abn*92	1
*abn*43	1	*abn*93	1
*abn*44	1	*abn*94	1
*abn*45	1	*abn*95	1
*abn*51	2	*nrm*01	1
*abn*52	2	*nrm*02	1
*abn*53	2	*nrm*03	1
*abn*54	2	*nrm*04	1
*abn*55	2	*nrm*05	1

The dendrogram shows that the normal group (*nrm*0*X*) is located at the lowest level. This means that normal event logs can be distinguished from faulty ones. Furthermore, there are groups of faulty event logs that are correctly classified at a certain level, such as *abn*2*X*, *abn*5*X* and *abn*8*X*. However, pairs *abn*1*X*−*abn*3*X*, *abn*4*X*−*abn*6*X*, and *abn*7*X*−*abn*9*X* are mixed. This is because faults in PU1 and faults in PU3 give the same observations, and similarly for pairs PU4 – PU6 and PU7 – PU9. In other words, pairs $O(G_{abn1})$ – $O(G_{abn3})$, $O(G_{abn4})$ – $O(G_{abn6})$, and $O(G_{abn7})$ – $O(G_{abn9})$ are indistinguishable, where O is the observation map used in the experiments.

The above result implies that if we give an event log w_{test} for a faulty situation, then the proposed method can identify a model from the following 7 groups:

- *nrm*0*X*
- *abn*1*X*, *abn*3*X*
- *abn*4*X*, *abn*6*X*
- *abn*7*X*, *abn*9*X*
- *abn*2*X*
- *abn*5*X*
- *abn*8*X*

6.5 Robustness for Values of Parameters

We examine robustness of the result with respect to values of the following three parameters:

1. The length l of each event log used for diagnosis.
2. The value of N.
3. The value of r.

We first prepare event logs consisting of different num-

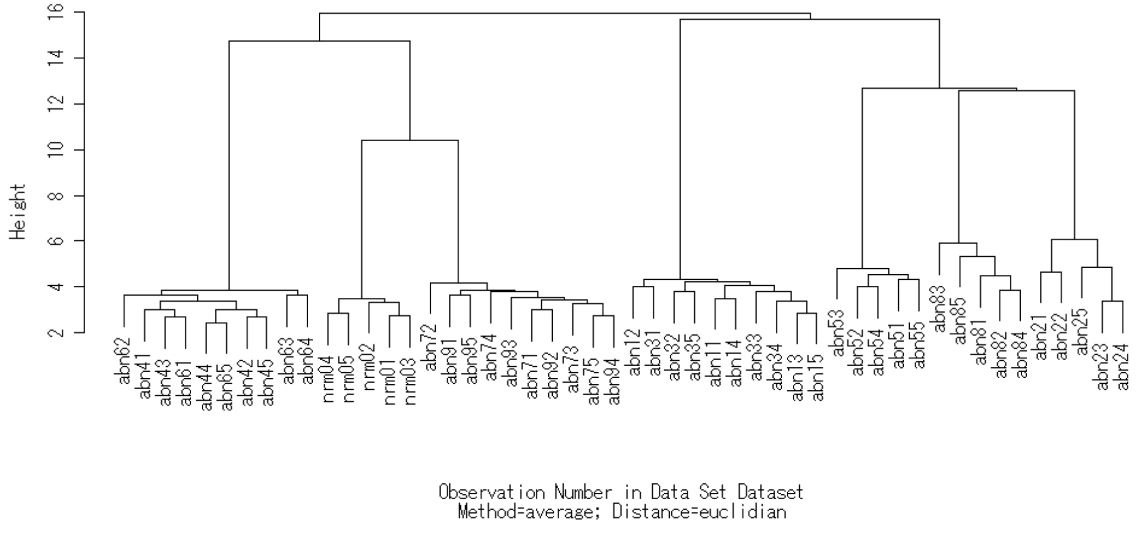

Fig. 6 Result of hierarchical clustering.

Table 2 Results for different log-length l ($r = 3$, $N = 2$).

$l =$	100	1,000	10,000	100,000
*nrm*1-*l*	1	1	1	1
*nrm*2-*l*	−0.0288	−0.0941	0.1428	0.1672
*abn*1-*l*	0.0132	0.04019	−0.1257	−0.1178
*abn*2-*l*	0.1214	0.2428	−0.0370	0.0599

Table 3 Results for different N ($r = 5$).

$N =$	1	2	3	4	5
*nrm*1	1	1	1	1	1
*nrm*2	0.7448	0.6065	0.4022	0.2055	0.2055
*abn*1	0.6069	0.4465	0.2566	0.0763	0.0763
*abn*2	0.5895	0.4534	0.2378	0.0949	0.0949

Table 4 Results for different r ($N = r - 1$).

$r =$	2	3	4	5
*nrm*1	1	1	1	1
*nrm*2	0.5203	0.2144	0.3019	0.2055
*abn*1	−0.2809	−0.0762	0.0684	0.0763
*abn*2	0.3583	0.0542	0.0071	0.0949

bers of events in the normal situation and two faulty situations. To check the separation of faulty situations from the normal situation, two event logs are generated from the normal situation. In total, there are sixteen event logs *nrm*1-*l*, *nrm*2-*l*, *abn*1-*l* and *abn*2-*l* for each length $l = 100,\ 1,000,\ 10,000$, and $100,000$. Then we compute correlation coefficients between the profile for *nrm*1-*l* and the profile of other logs, where we use $r = 3$ and $N = 2$. The result is shown in Table 2. It is observed that around 10,000 events is required for separating the normal situation from faulty situations. Of course, the necessary length of event logs may depend on the target system.

Next we prepare two logs *nrm*1 and *nrm*2 for the normal situation and two logs *abn*1 and *abn*2 for different faulty situations, where the length of each event log is 100,000. Then we compute the correlation coefficients between the profile of *nrm*1 and the profile of other logs for $r = 5$ and $N = 1, \cdots, 5$. The result is shown in Table 3. It is observed that $N = 4$ and $N = 5$ give almost the same result. Therefore, $N = r - 1$ might be a sufficiently large value for computing profiles.

Finally, we show results on different values of r, i.e., the length of the short sequences used in profiles. We prepare four event logs *nrm*1, *nrm*2, *abn*1 and *abn*2 consisting of 100,000 events, and compute correlation coefficients between the profile of *nrm*1 and the profile of other logs. The results are shown in Table 4. It is observed that the case for $r = 2$ does not give good separation but does in other cases. We need to choose an appropriate value for r. The optimal value of r depends on the structure of the probabilistic automaton representing the target system.

7. Conclusion

In this paper, we have presented a new method for diagnosis of discrete event system, called sequence profiling. Since order of events is taken into consideration in the N-gram model, the proposed approach is different from simple statistical analysis such as counting occurrences of each event. Short sequences used for the specificity correspond to local behavior of the system. When the target system is a distributed system like one used for the experiments, the effect of a faulty event is limited to the subsystem around the event, and other parts of the system runs normally. We expect that the vector of specificity for each event sequence reflects such behavior caused by local faults.

Whether faulty cases are distinguishable or not depends on the observation map. Designing optimal observation map for given purposes is one of future work. The results in Sect. 6 suggest that short event sequences do not give good separation. This is because the proposed method relies on differences in conditional probabilities. By this reason, the method shown in the paper is not suitable for detecting faults that immediately lead to system down. As demonstrated in the experiments, the proposed method is applicable to detecting non-functional faults such as a drop in performance at some part of the system. Diagnosis using shorter event logs also remains as future work.

Moreover, we need to improve the complexity in the calculation of the specificity for large N and r, in order to apply the method to larger and dynamically changing systems. In such systems, we should also consider how to deal with interleaved event sequences.

Acknowledgment

This research is partly supported by Grant-in-Aid for Scientific Research of MEXT under Grant No. 25330011.

References

[1] A.S. Willsky, "A survey of design methods for failure detection in dynamic systems," Automatica, vol.12, pp.601–611, 1976.
[2] P.M. Frank, "Fault diagnosis in dynamic systems using analytical and knowledge-based redundancy — A survey and some new results," Automatica, vol.26, no.3, pp.459–474, 1990.
[3] S. Rich and V. Venkatasubramanian, "Model-based reasoning in diagnostic expert systems for chemical process plants," Cornput. Chemn. Engng., vol.11, no.12, pp.111–122, 1987.
[4] M. Sampath, R. Sengupta, and S. Lafortune, "Diagnosability of discrete-event systems," IEEE Trans. Autom. Control, vol.40, no.9, pp.1555–1575, 1995.
[5] Xi Wang, I. Chattopadhyay, and A. Ray, "Probabilistic fault diagnosis in discrete event systems," 43rd IEEE Conference on Decision and Control, pp.14–17, 2004.
[6] D. Thorsley and D. Teneketzis, "Diagnosability of stochastic discrete-event systems," IEEE Trans. Autom. Control, vol.50, no.4, pp.476–492, 2005.
[7] W. Qiu and R. Kumar, "Decentralized failure diagnosis of discrete event systems," IEEE Trans. Syst. Man. Cybern. A, Syst. Humans, vol.36, no.2, pp.384–395, 2005.

[8] F. Liu, D. Qiu, H. Xing, and Z. Fan, "Decentralized diagnosis of stochastic discrete event systems," IEEE Trans. Autom. Control, vol.53, no.2, pp.535–546, 2008.
[9] A. Grastein, Anbulagan, J. Rintanen, and E. Kelareva, "Diagnosis of discrete-event systems using satisfiability algorithms," Proc. AAAI'07, vol.1, pp.305–310, 2007.
[10] G. Lamperti, M. Zanella, and P. Pogliano, "Diagnosis of active systems by automata-based reasoning techniques," Applied Intelligence, vol.12, no.3, pp.217–237, 2000.
[11] V. Chandola, A. Banerjee, and V. Kumar, "Anomaly detection: A survey," ACM Computing Surveys, vol.14, no.3, Article no.15, 2009.
[12] C.E. Shannon, "A mathematical theory of communication," Bell Syst. Tech. J., vol.27, pp.379–423, 623–656, 1948.
[13] M. Nagao and S. Mori, "A new method of N-gram statistics for large number of n and automatic extraction of words and phrases from large text data of Japanese," Proc. COLING'94, vol.1, pp.611–615, 1994.
[14] H. Ogawa, S. Inagaki, and T. Suzuki, "Decentralized fault diagnosis of event driven systems — Design of bayesian network between faults and events," Proc. SICE SSI2007, pp.327–332, 2007.
[15] W. van der Aalst, T. Weijters, and L. Maruster, "Workflow mining: Discovering process models from event logs," IEEE Trans. Knowl. Data Eng., vol.16, pp.1128–1142, 2004.
[16] V. Chandola, V. Mithal, and V. Kumar, "A comparative evaluation of anomaly detection techniques for sequence data," 8th IEEE International Conference on Data Mining, pp.743–748, 2008.
[17] D. Ron, Y. Singer, and N. Tishby, "The power of amnesia: Learning probabilistic automata with variable memory length," Machine Learning, vol.25, no.2-3, pp.117–149, 1996.
[18] C.C. Michael and A. Ghosh, "Two state-based approaches to program-based anomaly detection," Proc. 16th Annual Computer Security Applications Conference, p.21, 2000.
[19] P. Sun, S. Chawla, and B. Arunasalam, "Mining for outliers in sequential databases," SIAM International Conference on Data Mining, pp.94–105, 2006.
[20] W. Lee, S. Stolfo, and P. Chan, "Learning patterns from unix process execution traces for intrusion detection," Proc. AAAI Workshop on AI Methods in Fraud and Risk Management, pp.50–56, 1997.
[21] S. Forrest, C. Warrender, and B. Pearlmutter, "Detecting intrusions using system calls: Alternate data models," Proc. 1999 IEEE ISRSP, pp.133–145, 1999.
[22] http://www.informatik.uni-hamburg.de/TGI/PetriNets/tools/db/stpnplay.html

Miwa Yoshimoto received the B.E. degree in 1993 and the M.E. degree in 1995 from Kyoto Institute of Technology. She is studying systems science and interested in diagnosis of discrete event system at the postgraduate D.E. degree program at School of Information Science, Japan Advanced Institute of Science and Technology.

Koichi Kobayashi received the M.E. degree from Hosei University in 2000 and the D.E. degree from Tokyo Institute of Technology in 2007. He is currently an Assistant Professor at School of Information Science, Japan Advanced Institute of Science and Technology. His research interests include analysis and control of discrete event and hybrid systems. He is a member of the SICE, ISCIE, IEEJ, and IEEE.

Kunihiko Hiraishi received from the Tokyo Institute of Technology the B.E. degree in 1983, the M.E. degree in 1985, and D.E. degree in 1990. He is currently a professor at School of Information Science, Japan Advanced Institute of Science and Technology. His research interests include discrete event systems and formal verification. He is a member of the IEEE, IPSJ, and SICE.