An Approach of Data Mining Process Based on Stochastic Well-formed Workflows

Sha Jing and Yuyue Du
College of Information Science and Engineering,
Shandong University of Science and Technology, Qingdao, 266510, China

Abstract: As more and more event data become available, the practical relevance of data mining process is increasing. Process mining techniques aim to discover, monitor and improve real processes by extracting knowledge from event logs. A large volume of event data provides both opportunities and challenges for data mining process. The present process mining techniques have problems dealing with large event logs referring to many different activities. Therefore, we propose a generic approach to decompose process mining problems. It is possible to split computationally challenging process mining problems into many smaller problems that can be analyzed easily and whose results can be combined into solutions for the original problems. We present the matching algorithms to decompose the whole process model into several groups of traces and the numerical analysis of data mining models based on Stochastic Well-formed Workflow (SWWF).

Key words: Petri nets, process mining, stochastic well-formed workflow

INTRODUCTION

During the last two decades, there has been a shift from “data-aware” information systems to “process aware” information systems (Van der Aalst et al., 2004). To support business processes, an information system needs to be aware of these processes and their organizational context. Early examples of such systems were called Work Flow Management (WFM) systems (Dumas et al., 2005; Marinescu, 2002; Weske, 2007). In more recent years, vendors prefer the term Business Process Management (BPM) systems. BPM systems have a wider scope than the classical WFM systems and are not just focusing on process automation. BPM systems tend to provide more support for various forms of analysis (simulation) and management support (monitoring). Both WFM and BPM aim to support operational processes that are often referred to as “workflow processes” or simply “workflows”. In this study, we will use the generic term Process-Aware Information System (PAIS) to refer to systems that manage and execute such workflows.

DATA MINING PROCESS

Process mining is applicable to a wide range of systems. The only requirement is that the system produces event logs, thus recording (parts of) the actual behavior. For these event logs it is important that each event refers to a well-defined step in the process (a lab test) and is related to a particular case (a patient). Also, additional information such as the performer of the event (i.e., the doctor performing the test), the timestamp of the event, or data elements recorded along with the event (e.g., the age of the patient) may be stored. Based on these event logs, the goal of process mining is to extract process knowledge (e.g., process models) in order to discover, monitor and improve real processes.

Figure 1 positions process mining. Traditional data-oriented analysis approaches such as data mining (Hand et al., 2001) and machine-learning (Mitchell, 1997) do not consider processes, i.e., analysis focuses on particular decisions or patterns rather than the end-to-end processes. In contrast, Business Process Management (BPM) and Workflow Management (WFM) approaches focus on the analysis and improvement of end-to-end processes using knowledge from information technology and knowledge from management sciences.

Corresponding Author: Sha Jing, College of Information Science and Engineering, Shandong University of Science and Technology, Qingdao, 266510, China
Process models play a central role in BPM/WFM. Examples of process model analysis approaches are simulation (for what if analysis) and verification (to find design errors). As shown in Fig. 1, process mining combines both worlds to answer both performance and compliance related questions. The desire to link data and process is reflected by terms such as Business Process Intelligence (BPI) (Castellanos et al., 2004). However, only recently techniques and software have become available to systematically relate process models and event data (Van der Aalst, 2011). Industry reports such as (Manyika et al., 2011) and scientific studies (Hilbert and Lopez, 2011) describe the incredible growth of data. The term “big data” illustrates the spectacular growth of data and the potential economic value of such data in different industry sectors. Most of the data that are generated refer to events, e.g., transactions in some financial system and actions of some automated system.

The three most prominent process mining tasks are: (1) Process discovery: Learning a process model from example behavior recorded in an event log, inferring process models that are able to reproduce the observed behavior. Within the research domain of process mining, process discovery aims at constructing a process model as an abstract representation of an event log. The goal is to build a model (e.g., a Petri net, a BPMN model, or an EPC) that provides insight into the behavior captured in the log. For example, the discovered model may describe the typical steps taken before a surgery in a hospital. Note that also models describing the organizational, performance and data perspective may be discovered. (2) Conformance checking: Diagnosing and quantifying discrepancies between observed behavior and modeled behavior, checking if observed behavior in the event log conforms to a given model. For example, it may be checked whether a medical guideline which states that always a lab test and an X-ray needs to be done is always followed. (3) Extension: Projection of the information extracted from the log onto the model. For example, performance information may be projected on a discovered health-care process in order to see for which examinations a long waiting time exists.

**DATA MINING PROCESS BASED ON STOCHASTIC WELL-FORMED WORKFLOWS**

Petri nets are often used in the context of process mining. Various algorithms employ Petri nets as the internal representation used for process mining. Examples are the region-based process discovery techniques (Van der Aalst et al., 2016; Sole and Carmona, 2010), the algorithm (Van der Aalst and van Hee, 2004) and various conformance checking techniques (Adriansyah et al., 2011; Munoz-Gama and Carmona, 2011; Roizinat and van der Aalst, 2008). Other techniques use alternative internal representations (C-nets, heuristic nets, etc.) that can easily be converted to (labeled) petri nets (Murata, 1989; Van der Aalst, 2011). The process mining spectrum is quite broad and includes techniques like process discovery, conformance checking, model repair, roles discovery, bottleneck analysis, predicting the remaining flow time and recommending next steps.

We propose a method of decomposing a model into instance processes considering the dependencies among the instances.

In our network service environment, network service is transformed into SWWF-G. Large scale of network service is likely to consist of several SWWF-Gs. So, we only need to consider how to decompose one SWWF-G and how to find out the parallel network actions.

The concept of well-formed workflows (workflow built by atomic control blocks, nested control blocks and simple nodes) (Son et al., 2006) to avert structural errors is so rigorous that it will cause loss of flexibility in workflow schema design and make it difficult to describe complex schema. This study extends the well-formed workflow to include queuing network tasks patterns modeling the use of resource and execution of activities. The workflow net (WF-net) proposed by van der aalst is the high level petri nets with two special places i and o, which indicate the beginning and the end of the modeled process (Van der Aalst and van Hee, 2002).

In general, the WF-net is a marked graph. In the ideal case every transition is on a path and a fork and a join transition bound each path. A fork is a transition with more than one output places and a join is a transition with more than one input places. The WF-nets are suitable not only for representation and validation, but also for the verification of workflows. It was assumed that workflows had no structure errors when their performance was being computed and there were many approaches to verify a workflow (Van der Aalst, 2000; Sadiq and Orłowska, 1999).

**Definition 1:** (WF-net) A petri net $N = (P, T, F)$ is a WF-net (Workflow net) if and only if:

- There is one source place $s \in P$ such that $s = \phi$
- There is one sink place $o \in P$ such that $o = \phi$
- Every node $n \in P \cup T$ is on a path from $i$ to $o$

A WF-net does not care the concept of time, but sometimes we need to consider time aspect in workflow
management systems. For example, we want to know the completion time of a whole workflow net so that we can decide whether the arrangement of the workflow system meets our requirement for time and which activity is the bottleneck of the whole system. So, introducing time concept into WF-net is necessary.

In this study, we extend WF-net to SWWF-net by associating a poisson distributed arriving time and exponentially distributed serving time with each activity.

**Definition 2:** (SWWF-net) \( N_z = (P, T, r, F, \mu, \lambda) \) is a SWWF-net if and only if:

- \( N_z \) is structurally a WF-net
- The set \( T \) is a set of transitions denoting activities
- The set \( P \) is a set of places denoting states
- \( r = \{r_{ij} = [p_{b_1}, p_{b_2}, ..., p_{b_m}], m \in N, i = 1, ..., |P|, p_{b_j} \in R \} \) is a set of routing probability denoting the arc \((p_{b_i}, t) \in F \) \( p_{b_i} \in P, t \in T \)
- \( \mu = \{\mu_i = 1, ..., |T|, \bar{r} \in R \} \) is a set of arriving rates of activities
- \( \lambda = \{\lambda_i = 1, ..., |T|, \bar{r} \in R \} \) is a set of serving rates of activities

**ROUTING PATTERNS**

Since, the SWWF-net structurally inherits a WF-net, it also incorporates the four basic routing patterns proposed in Van der Aalst (2000): Sequential routing, parallel routing, selective routing and iterative routing, as shown by Fig. 2.

**Routing probability of an instance process:** An instance process represents a trace of workflow activities (transitions) that may be executed for a particular instance of a workflow. More exactly, the subset should include related arcs and places.

In general, a workflow consists of a set of instance processes having different routing probability by workflow cases. To automatically compute the routing probability of an instance sub-graph, we extend the notation of workflow nets by increasing routing probability to every arc pointing out from places. \( RP_{io} \) is used to denote the routing probability of the instance sub-graph in the following:

\[
RP_{io} = \prod_{i \in F_{io}} \left( \sum_{j=1}^{n_j} p_{b_j} \right)
\]  

(1)

The decomposition method has been applied in the selective pattern of the original model. Thus, Eq. 1 implies that the loop pattern doesn’t influence the routing probability of the instance processes.

**Response time of an activity:** As numerous natural physical and organic processes exhibit behavior that is probably meaningfully modeled by Poisson processes. An important application of the Poisson distribution arises in connection with the occurrence of events of a particular type over time. The exponential distribution is frequently used as a model for the distribution of times between the occurrences of successive events such as customers arriving at a service facility.

In this study, the Poisson process and the exponential distribution have been used to analyze many areas of computer engineering (Kleinrock, 1976). The Poisson process is used to model the arriving rate of activity instances and the exponential distribution is used to model the serving rate of activity instances.

From the queuing network theory, the queuing model is called a Markov queuing network if the input is a Poisson process and the serving time is an exponential distribution. The response time is used to obtain further results.

![Basic routing patterns of WF-net](image-url)

Fig. 2(a-d): Basic routing patterns of WF-net (a) Sequential, (b) Iterative, (c) Parallel and (d) Selective routing
For an activity $t_i$ having the arrival rate $\mu_i$ and serving rate $\lambda_i$, $\rho$ is the ratio between $\mu_i$ and $\lambda_i$. $R_i$ is used to denote the response time of $t_i$:

$$\rho = \frac{\mu_i}{\lambda_i}, \quad W_T = \frac{\rho}{\lambda_i(\rho - \rho^2)} = \frac{\mu_i}{\lambda_i(\rho - \mu_i)}$$  \hspace{1cm} (2)

$$R_i = W_T + S_T = \frac{\rho}{\lambda_i(\rho - \mu_i)} + \frac{1}{\lambda_i} = \frac{1}{\lambda_i}$$  \hspace{1cm} (3)

where, $W_T$ and $S_T$ denote the wait time and serving time of activity $t_i$, respectively. According to the queuing theory in operations research, the waiting time $W_T$ of a service is got through the Eq. 2. So, we get the response time of $t_i$ as Eq. 3 says.

**ALGORITHM**

Although, the different algorithms presented in the above section can handle many of the control-flow constructs, they are all unable to handle a common factor in real-life event logs: The presence of noise. Noise can appear in two situations: Event traces were somehow incorrectly logged (for instance, due to temporary system mis-configuration) or event traces reflect exceptional situations. In short, noise is any low-frequency behavior in a log. For instance, in our example of Fig. 1, one conference attendee that pays for parking and still travels by train would result in noise in the log.

The algorithm for computing the completion time of a whole SWWF model is showed in the following.

1. Decompose $cm$ into instance sub-graphs and put them into a set $I$
2. For each element $i$ in $I$, compute $R_P$ for $i$
3. Compute $T$ for $cm$

In the algorithm, the response time of each activity is computed in Step 2 according to Eq. 2. Step 3 is based on the algorithm proposed by Li et al. (2004). Step 4 computes the routing probability of each instance sub-graph by Eq. 1. The completion time of a whole model is computed in Step 10 and the formula is given as follows:

$$T_m = \sum_{i=1}^{n} R_P \times T_i$$  \hspace{1cm} (4)

where, $R_P$ and $T_i$ are the routing probability and response time of the $i$th instance sub-graph, respectively.

**CONCLUSION**

The concept of well-formed workflows is extended to stochastic well-formed workflows including queuing network activities in this study. Thus its ability of modeling complex business processes is enhanced. Also, an algorithm is proposed to analyze the performance of stochastic well-formed workflows. Owing to the ability of automatic computing routing probability, the time performance of each instance sub-graph and the performance of a whole workflow, our algorithm meet the frequent re-computing after business process reengineering.

**ACKNOWLEDGMENTS**

This study is supported by the National Natural Science Foundation of China under grants 61170078 and 61173042; the National Basic Research Program of China under grant 2010CB328101; the Doctoral Program of Higher Education of the Specialized Research Fund of China under grant 20113781100004; Basic Research Program of Qingdao City of China under grant 13-1-4-116-jch; the SDUST Research Fund of China under grant 2011KYTD102; the Research Project of “SUST Spring Bud” in Shandong University of Science and Technology in 2009; the Graduate Innovation Fund of Shandong University of Science and Technology; and the Shandong Province Higher Educational Science and Technology Program under Grant No.J13LNI8 And Qingdao Science Plan Application Basic Research Project under Grant No. 12-1-4-6-(9)-jch.

**REFERENCES**


