

On the discovery of process models from their instances[☆]

San-Yih Hwang*, Wan-Shiou Yang

Department of Information Management, National Sun Yat-Sen University, Kaohsiung, 80424 Taiwan

Accepted 30 November 2001

Abstract

A thorough understanding of the way in which existing business processes currently practice is essential from the perspectives of both process reengineering and workflow management. In this paper, we present a framework and algorithms that derive the underlying process model from past executions. The process model employs a directed graph for representing the control dependencies among activities and associates a Boolean function on each edge to indicate the condition under which the edge is to be enabled. By modeling the execution of an activity as an interval, we have developed an algorithm that derives the directed graph in a faster, more accurate manner. This algorithm is further enhanced with a noise handling mechanism to tolerate noise, which frequently occur in the real world. Experimental results show that the proposed algorithm outperforms the existing ones in terms of efficiency and quality. © 2002 Elsevier Science B.V. All rights reserved.

Keywords: Process discovery; Business process reengineering; Workflow management; Data mining

1. Introduction

The unprecedented growth of computer and networking technologies has changed the way in which enterprises operate. Organizations that seek to stay competitive in a rapidly changing environment are compelled to incorporate information technologies into many aspects of their business operations, which often call for the radical redesign of current business processes. Such a revolutionary change of business processes is termed *business process reengineering*, abbreviated as BPR. Reports on the successful implementations of BPR effort that achieve major improve-

ment on organizational objectives such as high service quality and low cost can be widely seen in the literature [12,20]. To perform BPR, several sets of guidelines have been proposed, including the five-step approach by Davenport [12], the six-step approach by Furey [13], and AT&T's seven-step approach [18]. Regardless of differences in their subtle details, these guidelines suggest that analysis of existing critical business processes as well as redesign of these processes are two essential BPR tasks. To facilitate these two tasks, a thorough understanding of the way in which the existing business processes currently practice is instrumental. Although organizations typically prescribe how business processes have to be performed, such prescription may not completely reflect the reality due to the following reasons:

- (1) Business processes are usually described in a loose manner such that many aspects are left

[☆] This work is supported in part by the National Science Council in Taiwan under grant number NSC89-2213-E-110-047.

* Corresponding author.

E-mail addresses: syhwang@mis.nsysu.edu.tw (S.-Y. Hwang), ryang@mis.nsysu.edu.tw (W.-S. Yang).

unspecified. This is especially common in human-oriented processes.

- (2) Some parts of the business processes are seldom executed and should be considered as exceptions thereafter.
- (3) The real processes are deviated from the pre-planned business processes because of environmental change.

One promising approach in improving process efficiency and customers' satisfaction, as advocated by many vendors and BPR experts, is to adopt a workflow management system (WFMS) that automates process executions [23]. A WFMS coordinates the execution of constituent activities as planned, enabling the tracking of ongoing process instances and reporting the statistical figures of processes being executed. However, current WFMSs assume that a precise model of all processes is available, whereas it has been widely recognized that defining a workflow type which totally represents all properties of the underlying business process is a difficult job [12]. Current practices for identifying a process model are usually performed in ad hoc manners, involving numerous meetings and discussions with authorized and knowledgeable persons.

Our primary objective in this paper is to propose a framework and develop algorithms for modeling the existing processes automatically. Specifically, we assume the existence of unstructured executions of a process, called instances. Taking the process instance data as the input, our algorithms will derive the control flow and the associated conditions of the underlying process. Instance data of a process may be collected in various ways. On one hand, in a traditional human-coordinated, document-driven process, instance data can be found in a collection of documents, each of which describes the execution information of a process instance, such as the completion time and the identity of the responsible person for each step involved. In this case, the discovered process model may help ease the introduction of a workflow management system. On the other hand, in an environment where a workflow system has been employed for coordinating process executions, detailed workflow logs are already available electronically (e.g., see Ref. [16] for a list of commercial WFMSs and the log information they provide). In this case, the discovered process model

serves as a feedback from the practical process executions and will help the evolution of the current process. In addition, commercial project management tools are also capable of recording some historical information about process executions. Some research prototypes have also been developed to monitor processes in a specific domain, such as software development [5].

1.1. Related work

The research on process discovery traces its origin to grammar discovery in the early 1970s [4]. The goal was to identify the underlying grammar from a finite number of sample strings. Grammars are commonly represented as finite state machines (FSMs). After a correct FSM is identified, it can then be used to tell the correctness of a given input string. More recently, researchers have started to adopt the existing grammar discovery algorithms to the problem of process discovery [9,11]. The idea was to treat an execution of a process as a string of events, each of which represents an execution outcome of an involving activity. With several executions of the same process as the input, these algorithms will be able to synthesize a process definition that best satisfies these historical data. Process definitions were described in the form of FSMs. For example, consider Fig. 1(a) for an example FSM of the program development process [9], which involves three sequential steps: code modification, compilation, and testing. After the code is modified (G), the subsequent compilation is performed and produces the result of either *OK* (I) or *not OK* (H). If the compilation is not okay, the code has to be modified again and the procedure has to be repeated; otherwise, a testing activity is performed. A successful testing (K) ends this process, and a failed testing (J) calls for the repetition of the entire procedure. Fig. 1(b) and (c) show the FSMs discovered from two different algorithms, KTAIL and Markov [9], respectively.

The original FSMs like the one shown in Fig. 1(a) are very easy to perceive and can be converted to an activity-based process model without much difficulty. As a matter of fact, in Fig. 1(a), each state corresponds to the execution of exactly one real-world activity, and its outgoing transitions represent the possible execution results. However, in a derived FSM, such as that in Fig. 1(b) or (c), a state may not have its clear semantic meaning, and an execution outcome may appear

متن کامل مقاله

دریافت فوری ←

ISIArticles

مرجع مقالات تخصصی ایران

- ✓ امکان دانلود نسخه تمام متن مقالات انگلیسی
- ✓ امکان دانلود نسخه ترجمه شده مقالات
- ✓ پذیرش سفارش ترجمه تخصصی
- ✓ امکان جستجو در آرشیو جامعی از صدها موضوع و هزاران مقاله
- ✓ امکان دانلود رایگان ۲ صفحه اول هر مقاله
- ✓ امکان پرداخت اینترنتی با کلیه کارت های عضو شتاب
- ✓ دانلود فوری مقاله پس از پرداخت آنلاین
- ✓ پشتیبانی کامل خرید با بهره مندی از سیستم هوشمند رهگیری سفارشات