Available online at www.sciencedirect.com
ScienceDirect
Information Systems 33 (2008) 64–95
www.elsevier.com/locate/infosys

Conformance checking of processes based on monitoring real behavior

A. Rozinat*, W.M.P. van der Aalst

Group of Information Systems, Eindhoven University of Technology, P.O. Box 513, NL-5600 MB, Eindhoven, The Netherlands

Received 28 April 2006; received in revised form 26 June 2007; accepted 1 July 2007
Recommended by B. Kemme

Abstract

Many companies have adopted Process-aware Information Systems (PAIS) to support their business processes in some form. On the one hand these systems typically log events (e.g., in transaction logs or audit trails) related to the actual business process executions. On the other hand explicit process models describing how the business process should (or is expected to) be executed are frequently available. Together with the data recorded in the log, this situation raises the interesting question "Do the model and the log *conform* to each other?". Conformance checking, also referred to as conformance analysis, aims at the detection of inconsistencies between a process model and its corresponding execution log, and their quantification by the formation of metrics. This paper proposes an incremental approach to check the conformance of a process model and an event log. First of all, the *fitness* between the log and the model is measured (i.e., "Does the observed process comply with the control flow specified by the process model?"). Second, the *appropriateness* of the model can be analyzed with respect to the log (i.e., "Does the model describe the observed process in a suitable way?"). Appropriateness can be evaluated from both a *structural* and a *behavioral* perspective. To operationalize the ideas presented in this paper a *Conformance Checker* has been implemented within the ProM framework, and it has been evaluated using artificial and real-life event logs.

Keywords: Business Process Intelligence; Process Mining; Petri Nets; Workflow Management

1. Introduction

New legislation such as the Sarbanes-Oxley (SOX) Act [1] and increased emphasis on corporate governance and operational efficiency have triggered the need for improved auditing systems. To audit an organization, business activities need to be monitored. Buzzwords such as BAM (Business Activity Monitoring), BOM (Business Operations Management), and BPI (Business Process Intelligence) illustrate the interest of vendors to support the monitoring and analysis of business activities. The close monitoring of processes can be seen as a second wave following the wave of business process modeling and simulation. In the first wave the emphasis was on constructing process models and analyzing them, illustrated by the many notations available

*Corresponding author. Tel.: +31 402475407; fax: +31 402432612.
E-mail addresses: a.rozinat@tue.nl (A. Rozinat), w.m.p.v.d.aalst@tue.nl (W.M.P. van der Aalst).

0306-4379/$ - see front matter © 2007 Elsevier B.V. All rights reserved.
doi:10.1016/j.is.2007.07.001

(e.g., Petri nets, UML activity diagrams, EPCs, IDEF, BPMN, and not to mention the vendor or system specific notations). This development has created the interesting situation where processes are being monitored while at the same time there are process models describing these processes. The focus of this paper is on *conformance*, i.e., "Is there a good match between the recorded events and the model?". A term that could be used in this context is "business alignment", i.e., are the real process (reflected by the log) and the process model (e.g., used to configure the system) aligned properly. Consider also Fig. 1, where conformance checking is positioned in the broader context of process mining techniques. While *discovery* aims at the automatic extraction of a process model from log data, conformance checking is concerned with the comparison of an existing process model and a corresponding log. As soon as one is confident in the conformance of the model and the log, *extension* techniques can be used to project diagnostic information derived from the log onto the model (for example, to visualize performance bottlenecks in the process model).

Most information systems, e.g., WFM, ERP, CRM, SCM, and B2B systems, provide some kind of *event log* (also referred to as transaction log or audit trail) [2]. Typically such an event log registers the start and/or completion of activities. Every event refers to a case (i.e., process instance) and an activity, and, in most systems, also a timestamp, a performer, and some additional data. In this paper, we only use the first two attributes of an event, i.e., the identity of the case and the name of the activity. Meanwhile, any organization documents its processes in some form. The reasons for making these process models are manifold. Process models are used for

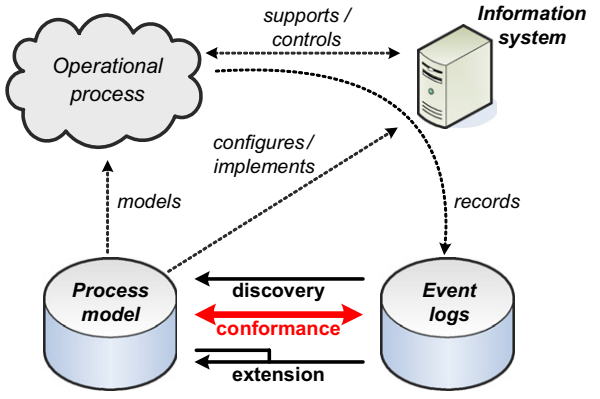

Fig. 1. Conformance checking in the broader context of process mining.

communication, ISO 9000 certification, system configuration, analysis, simulation, etc. A process model may be of a *descriptive* or of a *prescriptive* nature. Descriptive models try to capture existing processes without being normative. As an example, in a hospital process it must be possible to react to urgent situations and, therefore, the flexibility to diverge from the normal flow of actions is crucial. Another example could be a model that was made to document a certain procedure in a financial system (which logs events of the activities that were executed *without* being driven by an explicit process model). Clearly, it is desirable to keep this model aligned with the actual procedure in the financial system using regular conformance checking techniques. Prescriptive models describe the way that processes should be executed. In a Workflow Management (WFM) system prescriptive models are used to enforce a particular way of working using IT [3]. However, as shown in one of the case studies presented later in this paper, users may need to deviate even if they work with prescriptive models in a rigid WFM system. Furthermore, in most situations prescriptive models are not used directly by the information system. For example, the reference models in the context of SAP R/3 [4] and ARIS [5] describe the "preferred" way processes should be executed. People actually using SAP R/3 may deviate from these reference models. Finally, even if the process model and the event log are fully compliant, it is often interesting to see how frequent certain parts in the model are actually used, and to potentially remove obsolete parts which otherwise would need to be maintained.

In this paper, we will use Petri nets [6] to model processes. Although the metrics are based on the Petri net approach, the results of this paper can be applied to any modeling language that can be equipped with executable semantics. An event log is represented by a set of event sequences, also referred to as traces. Each case in the log refers to one sequence. The most dominant requirement for conformance is *fitness*. An event log and Petri net "fit" if the Petri net can generate each trace in the log. In other words: the Petri net should be able to "parse" every event sequence. We will show that it is possible to quantify fitness, e.g., an event log and Petri net may have a fitness of 0.66. Unfortunately, a good fitness does not imply conformance. As we will show, it is easy to construct Petri nets that are able to parse any event log. Although such Petri nets have a fitness of 1 they do not provide meaningful