Contents lists available at SciVerse ScienceDirect

Information Systems

journal homepage: www.elsevier.com/locate/infosys

Algorithms for anomaly detection of traces in logs of process aware information systems

Fábio Bezerra [a],*, Jacques Wainer [b]

[a] Cyberspace Institute – UFRA, Av. Presidente Tancredo Neves, 2501 Belém, Pará, Brazil
[b] Institute of Computing – UNICAMP, Av. Albert Einstein, 1251 Campinas, São Paulo, Brazil

ARTICLE INFO

Article history:
Received 9 March 2012
Received in revised form
24 April 2012
Accepted 25 April 2012
Recommended by D. Shasha
Available online 9 May 2012

Keywords:
Anomaly detection
Process mining
Process-aware systems

ABSTRACT

This paper discusses four algorithms for detecting anomalies in logs of process aware systems. One of the algorithms only marks as potential anomalies traces that are infrequent in the log. The other three algorithms: threshold, iterative and sampling are based on mining a process model from the log, or a subset of it. The algorithms were evaluated on a set of 1500 artificial logs, with different profiles on the number of anomalous traces and the number of times each anomalous traces was present in the log. The sampling algorithm proved to be the most effective solution. We also applied the algorithm to a real log, and compared the resulting detected anomalous traces with the ones detected by a different procedure that relies on manual choices.

1. Introduction and motivation

Process aware information systems (PAISs) are “a software system that manages and executes operational processes involving people, applications, and/or information sources on the basis of process models” [17]. In this paper we are interested in systems in which the execution of said processes are not predefined beforehand. Such systems fall within the ad hoc and loosely framed systems described in [17]. The central aspect of such loosely framed system is that the people that are executing the activities are in control to decide how the case will proceed next, in order to achieve the goals for the case. Such loosely framed system include flexible workflow systems, case handling systems, and scientific workflows.

These systems allow for more control by the people executing the process, and are a possible solution to what has been called the inflexibility of workflow systems [25,33]. But such flexibility may come with a cost. Systems that are not under control of a pre-specified process model may be subject to frauds and errors. Detecting such cases of frauds, exceptions and errors, which we will call *anomalies*, is the goal of this research.

From the point of view of this research, the execution of a case or an instance of a process is a sequence of activities that were executed on the behalf of that case. Thus the case “the firing of John Jacob Jingleheimer Schmidt” is an instance of a process of “firing”, and for Mr. Schmidt case the following activities were executed: “inform Mr. Schmidt”, “calculate balance due”, “explain severance benefits” and so on. In this paper, activities are considered atomic and their duration is not important, thus the set of activities executed can be seen as a *sequence*. Furthermore we will not attribute meaningful names to the activities, but refer to them using single letter names. Thus, Mr. Schmidt firing case is seen as the sequence of activities *abcbd*, for example. Such sequences of single letter activities are called *traces*. The set (or better the multiset) of traces from which one is trying to identify the anomalies is called a *log*. Each trace can appear many times in the log, and thus the multiset,

* Corresponding author. Tel.: +55 91 3228 1212.
E-mail addresses: fabio.bezerra@ufra.edu.br (F. Bezerra), wainer@ic.unicamp.br (J. Wainer).

0306-4379/$ - see front matter © 2012 Elsevier Ltd. All rights reserved.
http://dx.doi.org/10.1016/j.is.2012.04.004

and each time a particular trace appears in the log is called a *trance-instance*.

This research presents results in detecting anomalies in logs of execution of PAIS, where the anomaly is detected solely based on the sequence and choices of activities that took place in that anomalous execution. Thus, using the example above, one would detect that Mr. Schmidt firing was anomalous because the particular sequence *abcbd* of activities was too different from the sequences of activities for all or most of the other firing cases. For example, it may be the case that the activity "terminate Mr. Schmidt system access" was performed much later than usual, which could indicate either that the system administrator was not properly trained regarding the security policies, or that there was a collusion to allow Mr. Schmidt access to data he no longer should access.

Of course, the anomalous nature of a case may be derived from the values involved in some of the activities (for example, Mr. Schmidt's health benefits remain active for 300 month after his firing), or because of the people who executed some of the activities (for example, the system access termination activity was executed by a senior vice president), or because of time to perform an activity or the whole process was greater or less than normal (for example, the calculation of balance due was faster then normal). We call these examples as *data*, *organizational*, and *time* anomalies, to match the other four aspects of process models [36]. This research is restricted to *control flow* anomalies.

1.1. Toward a definition of anomalous trace

Anomalous traces, once discovered, must be analyzed to find out if indeed they are examples of incorrect executions or if they are acceptable executions, and if they are found to be incorrect executions, the reasons for and consequences of these executions must be further investigated. Thus, the algorithms discussed in this paper must be used as a first automated step toward a more comprehensive security auditing practice for flexible or loosely framed PAIS. This places some constraints for the algorithms. If we focus on a fraud perspective—that is, that the anomalous traces are a possible indication of frauds, then missing any of the potential fraudulent executions has serious consequences. Thus, the algorithms to detect the anomalous traces must have a very low false negative rate. The false negative cases are traces that the algorithm flagged as negative (or "normal") and that attribution was wrong. Such false negative cases will not be forwarded to the specialists that would determine that the trace was indeed a fraud and take the appropriate measures. On the other hand, given that this human analysis of whether an anomalous trace is indeed a fraud is a costly process, one would also prefer if the algorithms had low false positive rates – that is, the number of cases that are mistakenly flagged as anomalous when in fact they are not – should also be kept low. But a low false positive rate is less important than a low false negative rate.

If the anomalous traces are interpreted as errors, either erroneous execution or erroneous logging of the processes, then the unbalance of costs between a false negative and a false positive is less severe. A false negative will not generate the loss of revenue that an undetected fraud usually will incur. Therefore under this perspective a more even balance between false negative and false positive rates should be aimed at. In this paper we will also explore this alternative.

Finally, let us address the issue of what is an anomalous execution of a process. Chandola et al. [10], in an important survey on anomaly detection, discuss that there is no formal definition of anomaly, only intuitions that guide the development of different algorithms and techniques. For example, one may have the intuition that "normal" data falls "together" (in some appropriate distance metric) and that anomalies are "spread apart". This intuition based on distance leads to the development of many algorithms based on nearest neighbor [10, Section 5]. If on the other hand, one has the intuition that anomalies are data points that have low probability of occurring (given the appropriate generative model for the "normal" data), this intuition leads to the development of family of techniques described in [10, Section 7] as statistical detection models.

The same apply to our research: we have no formal definition of an anomalous traces, but we have some intuitions that guided the development of the algorithms discussed herein. They are

- the set of executions can be partitioned into a set of *normal* and *anomalous* executions,
- each of the anomalous execution is "infrequent" among the set of all executions, although the whole set of anomalous executions may not be that infrequent,
- the process models that "explain" the executions in the normal set "make sense",
- the process models that could explain both the normal executions and some of the anomalous ones "make less sense".

The terms "infrequent", "explain" and "make sense" need to be further refined if one wants to transform these intuitions into one or more algorithms. Nevertheless these intuitions can be formalized in some more precise notation, leaving the uncertainties confined into a few constants and relations

- given a set A of activity names,
- a trace t is defined as $t \in A^*$,
- a log L is defined as a multiset of traces $L = \{\langle t, n_t \rangle\}$ where n_t is the multiplicity of the trace in the log,
- the size of a log L is the number of trace-instances in it, that is $\forall \langle t, n_t \rangle \in L$, $\text{size}(L) = \sum n_t$,
- the frequency of a trace t in the log L is defined as $\text{freq}_L(t) = n_t / \text{size}(L)$,
- there exists a constant freq_{max} represents the term "infrequent",
- there exists a relation "explain" between a process model M and a log L denoted by $M \vdash L$,