Available online at www.sciencedirect.com
SCIENCE DIRECT
COMPUTERS IN INDUSTRY
Computers in Industry 53 (2004) 283–296
www.elsevier.com/locate/compind

# Discovering workflow models from activities' lifespans

Shlomit S. Pinter*, Mati Golani
*IBM Research Laboratory in Haifa, Haifa 31905, Israel*

**Abstract**

Workflow systems utilize a process model for managing business processes. The model is typically a directed graph annotated with activity names. We view the execution of an activity as a time interval, and present two new algorithms for synthesizing process models from sets of systems' executions (audit log). A model graph generated by each of the algorithms for a process, captures all its executions and dependencies that are present in the log, and preserves existing parallelism.

We compare the model graphs synthesized by our algorithms to those of Agrawal et al. [Mining process models from workflow logs, in: Proceedings of the Advances in Database Technology (EDBT'98), 6th International Conference on Extending Database Technology, Valencia, Spain, 23–27 March 1998, Lecture Notes in Computer Science, Proceedings vol. 1377, Springer, Berlin, 1998] by running them on simulated data. We observe that our graphs are more faithful in the sense that the number of excess and absent edges is consistently smaller and it depends on the size and quality of the log. In other words, we show that our time interval approach permits reconstruction of more accurate workflow model graphs from a log.


*Keywords:* Workflow mining; Process discovery; Event log; Activity lifespan

## 1. Introduction

Constructing business processes is a central issue for companies [8,20]. Managing processes in an automatic or semi-automatic fashion result in a significant reduction of cost and improves efficiency of business operations, thus, enabling fast adaptation to changing requirements and more. As a result, developing techniques for constructing and managing business processes is an active research area [2,6].

Workflow systems utilize a visual model of information flow that is used for monitoring and managing systems that execute actions (also called activities or tasks) of predefined situations. The actions together with constraints on execution order between them define the business process [11]. Commercial workflow systems and management consoles need a model of the business process for scheduling agents (e.g., computers) to execute the actions, control production, etc. (see [11,8]). For modeling a business process, most ERP/CRM products use embedded workflow model [7,19].

Many organizations that run their systems using legacy applications do not have a model of the processes within the organizations. Current tools for model detection operate on the resource level only. Thus, there is a need for tools to build business process level models, especially when all executive level measures such as "return on investment", or SLA quality are derived from this level rather than the resource level. There are few methods for constructing a business process model from information stored in a workflow log (a collection of process execution logs). We follow the approach that represents the model as a directed

* Corresponding author. Tel.: +972-4-829-6541; fax: +972-4-829-6116.
*E-mail address:* shlomit@il.ibm.com (S.S. Pinter).

0166-3615/$ – see front matter 
doi:10.1016/j.compind.2003.10.004

graph (workflow graph) with nodes representing activities and an edge from node A to node B represents that there is a process execution in which A must finish executing before B starts. In practice, a single business process model can permit an execution that include a given activity and another execution that may not include it. Thus, for each process execution the participating edges are selected with a Boolean function associated with each edge. The function determines whether the control flow or not along that edge.

Another paradigm is to deal with workflow evolution that updates process models according to the logs [1,3,7,12,13,15,17].

### 1.1. Contribution

Unlike the event based models in which the execution of an activity is represented as a single event, we view the execution of an activity as a time interval (life span) based on its starting and ending events that are present in the workflow log. For this view we can recognize concurrent activities with a single process execution, since intersecting life span intervals in a process execution represent concurrent activities. In event based models concurrent activities cannot be recognized in a single process execution. We present a new algorithm based on our interval model. Given a workflow log we show that compared with the event based approach, the interval approach enables the reconstruction of a more accurate process model graph.

We define causality dependence between activities with respect to the process execution log and describe an algorithm that, given a process execution log, generates a workflow graph that guarantees the following:

- *Completeness:* Every process execution in the log can be generated from the workflow graph.
- *Correctness:* All the dependencies with respect to the execution log exist in the workflow graph.
- *Preserving parallelism:* If two activities, A and B, are concurrent with respect to the log then there are two paths from start to end such that one includes A and does not include B and the other includes B and does not include A.

A workflow graph generated by our algorithm resemble the original workflow graph (the input for the system that generated the log) by having a minimum number of excess and absent edges. This minimum depends on the size and quality of the log. The time complexity of the algorithm is $O(|ex||V|^2)$, where $|ex|$, $|V|$ are the number of executions and activities in the log, respectively. We measure the quality of the algorithms by computing their *precision* and *recall* values, where precision is the ratio of correctly identified edges over the total number of generated edges, and recall is the ratio of correctly identified edges over the total number of edges in the original workflow graph (the targeted workflow graph).

In their seminal paper [2], the authors build a workflow graph by considering each activity as an atomic event. This was achieved by taking the finish activity event to be a node in the graph. We show that when the interval view is taken there is more information that can be used for reconstructing a more accurate model. We compare our results with those of [2] and show that the quality of the generated models is better with our algorithm.

### 1.2. Background and related work

The topic of process discovery has been dealt for some time [4–6,9,10,21]. Our work can be viewed as an extension to the event based approach taken by [2,9]. They use a single event to denote the execution of an activity and reconstruct a directed acyclic graphs as the model. The common approach as described in those studies was to identify the different business processes in the execution log and gather the participating activities in each execution.

In [5,6], Cook and Wolf searched for software processes by investigating events in concurrent environment. They present three methods for process discovery: a neural networks based method, algorithmic approach which builds a finite state machine where states are fused if their futures (in terms of possible behavior in the next k steps) are identical, and a Markovian approach which uses a mixture of algorithmic and statistical methods and takes noise in consideration. The Authors propose specific metrics (entropy, event type counts, periodicity, and causality) and use these metrics to discover models out of event streams. However, they do not provide an approach to generate explicit process models. In [4], they provide a measure to quantify discrepancies between a process