

Mining Process Knowledge – Algorithms and Experience with Simulated and Real Logs

Akhil Kumar and Zan Huang

Pennsylvania State University

Recently, there has been considerable interest in extracting process models from logs of actual execution instances, which is often referred to as *process mining* (Aalst et al. 2003; Dustdar et al. 2005). The goal of process mining is to extract semantic knowledge for the purposes of process understanding, innovation and improvement. It has many potential applications in the real world in any area where processes arise, such as business, healthcare, and even scientific fields.

Consider a simple log containing two process instances (also referred to as rows): { ‘A B C D’, ‘A C B D’ }, consisting of four tasks, A through D. Each process instance corresponds to the actual execution order of a workflow process. The first instance suggests that the four tasks were executed sequentially. However, after looking at the second instance it appears that B and C have a parallel relationship between them. Since a log is always written sequentially, parallelism of two tasks (or activities) has to be inferred from knowing that the two tasks can appear in any sequence. On the other hand, if two tasks always appear in a strict order with respect to one another, this suggests there is sequential relationship between them. Next, let us say a third instance ‘A B C F B C D’ were also present in the log. Now, this suggests that there is a loop in which tasks B and C are repeated. It also tells us that there is another task F. Thus, one can see that all the information in the log must be combined to derive a process that actually conforms to all the instances.

This is a hard problem because one is taking a sequential log representing execution traces, and trying to reconstruct a semantic model from it. There are many open research issues on this problem, such as:

- What is considered a complete log or a partial log?
- How can a process model be extracted from a complete log?
- What information can be derived from a partially complete log?
- How can we derive a process model if the log is noisy?
- What is the best way to present a complex process model to the user?

We propose a new algorithm to extract a *block* structured model of a workflow process from a log consisting of actual process execution sequences. The algorithm considers four basic block structures, the sequence, choice, parallel and loop structures, as building blocks of the model. The algorithm uses a unique approach of creating two kinds of relationship matrices, direct and indirect, between all pairs of tasks to capture the relationship patterns embodied in the log. Then it looks for patterns in these matrices to deduce an underlying structure between the tasks of the process. This procedure is repeated until the complete workflow is discovered. In order to accommodate real-world logs with noises we also developed preprocessing techniques to handle special structures like one-loops and optional or missing tasks.

Although graphical and Petri net representations of workflow models are useful, the advantage of block structured models is that they are easier for business analysts and end users to understand. Further, there are several tools and languages (including WS-BPEL) that primarily support block structured workflows.

We are in the process of testing the proposed algorithm using simulated logs from randomly generated process models and conducting an extensive experiment with and a relatively large real data set.

References

Aalst W.M.P van der, B.F. van Dongen, J. Herbst, L. Maruster, G. Schimm, A.J.M.M Weijters:
"*Workflow mining: A survey of issues and approaches*". DKE 47 (2003).

Dustdar S., T. Hoffmann, and W.M.P. van der Aalst. Mining of ad-hoc business processes with TeamLog. *Data and Knowledge Engineering*, 55(2):129-158, 2005.