New Quality Metrics for Evaluating Process Models

Zan Huang and Akhil Kumar

Smeal College of Business, Penn State University, University Park, PA 16802, USA
{zanhuang,akhilkumar}@psu.edu

Abstract. In the context of business process intelligence, along with the need to extract a process model from a log, there is also the need to measure the quality of the extracted process model. Hence, process model quality notions and metrics are required. We present a systematic approach for developing quality metrics for block structured process models, which offer less expressive power than Petri-nets but have easier semantics. The metrics are based on tagging an initial block structured process model with self-loop and optional markings in order to explain all the instances in the given log. Then we transform the marked model to an equivalent maximal model by rewriting the self-loop and optional markings for consistency, and determine a badness score for it, which determines quality. Our approach is compared with related work, and a plan for testing and validation on noise-free and noisy data is discussed.

Keywords: Process mining, process logs, Petri-nets, block structured models, badness score, quality-metric, equivalent models, self-loops, optional tasks, noisy log.

1 Introduction

Recently, there has been increasing interest in extracting process models from logs [1-4]. It has also been noted that in general many different process models can be extracted from the same log. This has given rise to the issue of process model quality with a need for being able to distinguish good models from bad models [7]. Ideally, a good model is one that produces *high fidelity* or *fitness* [6], i.e. explains all the log instances in a given log, along with *specificity*, or behavioral appropriateness [6], i.e. it only explains the given log and no other log. It should also be minimal [8]. In this paper, we consider logs generated from block structured process models. These models are composed by combining four basic control structures, sequence, parallel, choice and loop, and two additional structures proposed in this paper, the optional and self-loop structures. We develop three quality metrics for evaluating block structured process models and show how they can be applied to determine process model quality. We also discuss the issue of noise in a log in the context of these metrics.

2 Preliminaries

A *block structured process* is created by using four basic structures as building blocks: *sequence, parallel, choice* and *loop*. These are shown in Figures 1(a)-(d). In a block structured model atomic tasks are combined into sub-blocks, which are further combined into larger sub-blocks, using one of these four basic structures, until the full model is created. If a log corresponds exactly to such a block structured model, then one can limit oneself to just these four structures and discover a model from a log. However, in real logs that we observed, we found that it is seldom the case that a log follows this basic model perfectly. For example, an entry like AABCBC cannot be described using the four basic primitives of the block structured model. In order to capture these two scenarios two additional constructs – *optional* and *self loop* (respectively) structures are introduced. An optional structure (see Figure 1(e)) is a choice with one empty branch. A self loop (Figure 1(f)) is a loop of a single task/block. These can create severe complications in the discovery process. In the remaining discussion we use the terms *task* for atomic, lowest level tasks, and the term *sub-block or block* for either an individual task or an aggregate of two or more tasks created by using one of the structures of Figure 1.

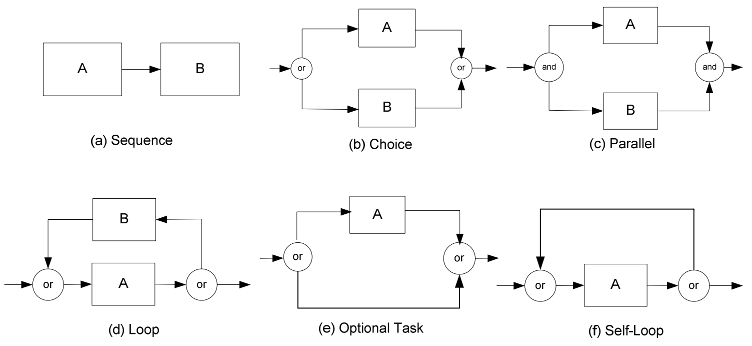


Fig. 1. Basic Types (or Patterns) of Block Structured Workflows

An important implication of introducing the optional and self-loop structures is that models can be constructed to describe every possible log. We refer to these models as *universal models*. Any given model can be converted into a universal model by adding self-loop and optional tags. However, there is a major problem with the universal model. The *quality* of this model is *bad* since it can simulate *every* log. Thus, there is no way to distinguish between a process model for a given log versus the process model for another log. Hence, this model lacks *specificity*.

Here we develop some axioms to capture the properties of a quality metric.

Axiom 1: Quality measures both fidelity and specificity of a process model to a log.

Axiom 2: A universal model that can describe **every log** has the worst quality.

Axiom 3: All universal models should have equal quality score, i.e. the worst.

Axiom 4: Between two models with the same number of tasks, the one that allows more traces should have a lower quality value than the other one.

These axioms will apply to our quality metric developed in Section 4, but first we review our algorithm for generating a model from a log.

3 An Approach for Process Model Quality Metrics

3.1 Assigning Optional and Self-loop Structures

Given an initial block-structured model of four basic structures from a given log, which can be generated using algorithms such as the backtracking algorithm [5], Petri Net algorithm in the ProM framework [1], and genetic algorithms [2], we *fine-tune* the model using a **replay** function. The replay function takes a proposed model as an input and walks through each log instance to assign self-loop and optional labels as needed in order to ensure that the log instance agrees with the model. For each log instance, the algorithm creates an initial set of *next_tasks*, at the start of the process, and recomputes this set after any task corresponding to an entry in the log is completed. Moreover, we also keep track of tasks that have been done in a set called *done_tasks*. As each successive entry *e* in a log instance is scanned there are three cases: If *e* is found in the set of *next_tasks*, then the set of next_tasks is recomputed for the next entry *e* in the log instance; If it is not found in the set of *next_tasks*, but it is in the set of *done_tasks*, then it or its parent is part of a self-loop block since it is already done and is repeating unexpectedly; If it is in neither set, this means that several expected *next_tasks* have been skipped and should be marked 'optional'. This is done by finding all possible *paths* in the process between the tasks in the set of *next_tasks* and *e*, and adding all tasks on these paths to the set of *skipped_tasks*. Thus the scan of each log instance results in two 0-1 bit-vectors of size 2n-1 each to track whether a task or block is a self-loop or optional. The union of these bit-vectors gives a model that agrees with all the log instances. A full listing of **replay** appears in [5].

3.2 Model Equivalence

Since *any* process model can be made to conform 100% to any given log, to evaluate and compare process models, we need quality metrics. The quality metric must be the same for equivalent models hence we first discuss the notion of model equivalence. We generate all possible log instances of a model of a single block consisting of two tasks from all 256 combinations of values for structure (4 types) and SL and OP bit-vector values (64 combinations). The models that generated identical set of log instances (with maximum log length limited to 6 to allow for sufficient variation of log instances for different structures) were marked as **equivalent** and stored in a lookup table. Among each set of equivalent combinations we also identified minimal (maximal) combinations as ones with the smallest (largest) number of 1 bits. For example, the loop structure in Figure 1(d) with A as optional and the same structure with A and the block as optional and B as self-loop are the minimum and maximum models of a set of equivalent models. With the equivalence lookup table for the two-task models, we can then rewrite a given process model (with more than 2 tasks) into its equivalent maximum model, which then is used to determine the badness score and quality metric. Because the rewriting can cause the child tags to affect the parent tags and vice-versa, we iterate until convergence is achieved, i.e. there is no change in successive iterations.

3.3 Badness Score

Our *quality metric* relies on a *badness score*. This score measures the badness of a model where, intuitively, a model with more self-loops and optional tasks and blocks is "more bad" than one with fewer of them. We use a recursive formula as follows:

$$B(n)=\begin{cases}(1+W1.op(n)+\ W2.sl(n)).(B(n1)+B(n2)), & n\text{ is a block}\\ 1+W1.op(n)+W2.sl(n), & n\text{ is a task}\end{cases}$$

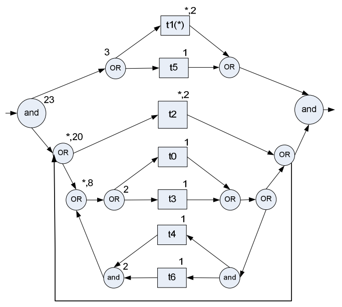
where $n1$, $n2$ are child nodes of *n, op*() and *sl*() are 0-1 functions to indicate whether a given block is of optional and self-loop structure. If we think of a process as a tree the B value of the root will give the badness of the process. W1 and W2 are weighting factors to vary the influence of self-loops and optional tasks/blocks. For now they are set to 1 for illustration, while empirical tests are done to find optimal values for them.

3.4 Examples Models and Badness Scores

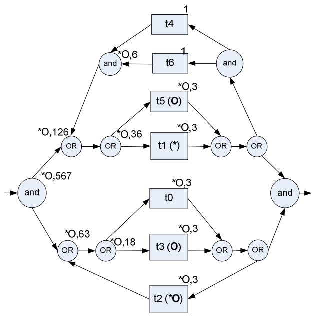
Our example is based on the simple log of Figure 2(a) with 7 tasks. Using a backtracking algorithm we generated three models for this log, as shown in Figures 2(b)-(d). In these models the structures are generated first, and then the self-loop (*) and optional (O) tags are added in the task box by scanning each log instance and trying to make the model consistent by replaying the log. A marking of *O after a task means that the task is both in a self-loop and is optional. For a block a self loop is shown by a directed link from the

t1 t3 t9 t5 t3 t9 t5 t0
t1 t3 t9 t5 t0 t2
t0 t2 t2 t8
t1 t0 t1
t3 t8 t5 t9 t0 t9 t5 t3

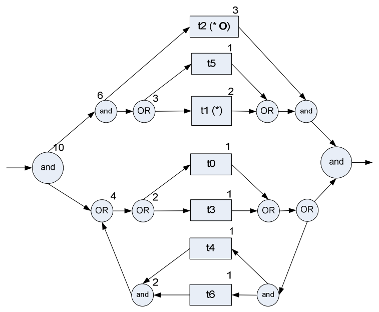
(a) An example log with five instances of execution traces



(b) Model PM1, badness score = 23



(c) Model PM2, badness score = 567



(d) Model PM1, badness score = 10

Fig. 2. Process model PM1 and PM2 generated by our algorithm

join node to the split node of the block. It is possible to verify that after adding the self-loop and optional markings, all models are consistent with the given log. We first performed rewriting for the entire model as explained in Section 3.2 to obtain the equivalent maximal model, and then determined a badness score. The self-loop and optional markings after rewriting are shown above tasks and blocks, together with the badness scores associated with the blocks/task. An atomic task contributes 1 to the score. However, if it has a self-loop or optional tag, then each tag adds 1 to the score. The initial score of a parent node is equal to the sum of the child node scores. Moreover, if a parent has a self-loop or optional tag on it, then each tag will result in the initial score being added yet again.

3.5 Quality Metrics

In this section we introduce three quality metrics. A simple metric Q0 is based on counting the number of self-loops and optional structures in a maximally equivalent process model of a given model as follows (where T is the number of tasks):

$$Q0 = 1 - \text{(Number of self loops + number of optional markings)}/(4 * T - 2)$$

However, the Q0 metric does not reflect the level at which a marking appears. Clearly a self-loop or optional marking at a higher level can create a lot more paths in the process, and hence hurt the quality more than such a marking at a lower level. Therefore we consider a new metric based on the badness score (B-score):

$$Q1 = \text{Log(T)/Log(Bscore of actual model)}$$

The numerator represents the badness score for the best model which is T since it does not have any markings. A third metric Q2 is as follows:

$$Q2 = 1 - \text{Log(Bscore of actual model} - T + 1\text{)/Log(Bscore of universal model} - T + 1\text{)}$$

The badness score of the universal model is calculated assuming that each task and block has both self-loop and optional markings. This score has the property that it is the same for all universal models. Thus, for T = 7, B-score-universal (7) = 5463.

All three metrics are based on the self-loop and optional markings of the initial block-structured model. The more such markings we have to add to a given initial model, the more log instances are inconsistent with the initial model (reflecting fidelity of the model) and the more general the tagged model becomes (reflecting specificity of the model). Thus, these metrics do reflect both fidelity and specificity of a given model. For models with the same number of tasks, the universal models all have the worst (and equal) values for all three metrics. Q0 and Q2 are always zero for universal models, while with a larger number of tasks (say, more than 10) Q1 is also a small number for universal models and it gets close to zero as the number of tasks increases.

To do a preliminary validation for Axiom 4, we compared the number of log instances of models consisting of only 2 tasks (with the maximum length of an instance set to 6) and the quality metrics of these models. The Spearman rank order correlation coefficients for Q0, Q1, and Q2 are –0.7659, –0.75, and –0.75, respectively, all with *p-values* smaller than 0.001, indicating statistical significance of the association between all three quality metrics and number of possible log instances. The correlation coefficient for Q1 and Q2 are identical as both are monotonic transformations of the badness score. The correlation coefficient is high but not perfect for two reasons. First, the metrics are approximate. Secondly, they do not consider the effect of the four basic structures on the number of path sequences (e.g., a parallel structure creates more paths than a sequence). Finally, our experiments also show that fine tuning the weights W1 and W2 can increase correlation even more. We are conducting experiments to optimize W1 and W2.

The three models PM1, PM2, and PM3 are ranked in the same order by the three metrics which is consistent with the axioms, but the ranges are different (see Table 1). The range for Q0 is the largest and for Q2 it is smallest. Q0 is somewhat simplistic because it is based on a count of self-loops and optional structures. Q1 and Q2 are more accurate because they rely on the B-score. Nevertheless, the drawback with Q1 is that it gives a value of 0.22 for the universal model, so the range of quality is from 0.22 to 1. Of course, as the number of tasks increases, the minimum value of Q1 also drops, yet Q2 is a better metric since it covers the full range from the best to the worst model. On the other hand, Q1 offers a relative comparison with the best model.

Table 1. Calculations of the three metrics for various models

	PM1	PM2	PM3	Universal	Best
Q0	0.846	0.154	0.885	0	1
Q1	0.621	0.307	0.842	0.22	1
Q2	0.678	0.280	0.739	0	1

4 Conclusions and Future Work

Block structured process models have lesser expressive power compared to Petri-net based workflow models, but they offer a simpler semantics from an end-user point of view. Therefore, it is useful to study process quality issues in the context of such models as well. We presented a systematic approach for developing quality metrics for block structured process models and demonstrated the calculation of these metrics. The metrics are based on creating an initial model and then tagging it with self-loop and optional markings in order to explain all the instances in the log. We transformed the marked model to an equivalent maximal model by rewriting, and determined a badness score, which is used to calculate quality. The use of the metrics was illustrated with an example. The results from testing and validation on simulated and real data are provided in [5]. We also plan to explore ways to increase the expressiveness of our models by allowing additional structures.

References

1. van der Aalst, W.M.P., van Dongen, B.F., Günther, C.W., Mans, R.S., Alves de Medeiros, A.K., Rozinat, A., Rubin, V., Song, M., Weijters, A.J.M.M., Verbeek, H.M.W.: ProM 4.0: Comprehensive support for real process analysis. In: Kleijn, J., Yakovlev, A. (eds.) ICATPN 2007. LNCS, vol. 4546, pp. 484–494. Springer, Heidelberg (2007)

2. van der Aalst, W.M.P., de Medeiros, A.K.A., Weijters, A.J.M.M.: Genetic process mining: an experimental evaluation. Data Mining and Knowledge Discovery 14(2), 245–304 (2007)
3. van der Aalst, W.M.P., van Dongen, B.F., Herbst, J., Maruster, L., Schimm, G., Weijters, A.J.M.M.: Workflow mining: A survey of issues and approaches. Data and Knowledge Engineering 47(2), 237–267 (2003)
4. van der Aalst, W.M.P., Weijters, A.J.M.M., Maruster, L.: Workflow mining: Discovering process models from event logs. IEEE Transactions on Knowledge and Data Engineering 16(9), 1128–1142 (2004)
5. Huang, Z., Kumar, A.: A study of process mining: Quality and accuracy trade-offs. Smeal Working Paper, Pennsylvania State University (September 2008)
6. Rozinat, A., van der Aalst, W.M.P.: Conformance checking of processes based on monitoring real behavior. Information Systems 33(1), 64–95 (2008)
7. Rozinat, A., de Medeiros, A.K.A., Günther, C.W., Weijters, A.J.M.M., van der Aalst, W.M.P.: The need for a process mining evaluation framework in research and practice. In: Proceedings of the 3rd Workshop on Business Process Intelligence, pp. 84–89 (2007)
8. Schimm, G.: Mining most specific workflow models from event-based data. In: van der Aalst, W.M.P., ter Hofstede, A.H.M., Weske, M. (eds.) BPM 2003. LNCS, vol. 2678, pp. 25–40. Springer, Heidelberg (2003)