Business Process Quality Metrics: Log-based Complexity of Workflow Patterns

Jorge Cardoso

Department of Mathematics and Engineering,
University of Madeira, Funchal, Portugal
jcardoso@uma.pt

Abstract. We believe that analysis tools for BPM should provide other analytical capabilities besides verification. Namely, they should provide mechanisms to analyze the complexity of workflows. High complexity in workflows may result in poor understandability, errors, defects, and exceptions leading processes to need more time to develop, test, and maintain. Therefore, excessive complexity should be avoided. The major goal of this paper is to describe a quality metric to analyze the complexity of workflow patterns from a log-based perspective.

Keywords: workflow, process log, workflow complexity, business process quality metrics, business process analysis.

1 Introduction

Workflow verification tools such as Woflan [1] are indispensable for the current generation of WfMS. Yet, another desirable category of tools that allows building better workflows are tools that implement workflow quality metrics. In the area of software engineering, quality metrics have shown their importance for good programming practices and software designs. Since there are strong similarities between software programs and business process designs, several researchers have recognized the potential of quality metrics in business process management [2-5].

In [6], Vanderfeesten et al. suggest that quality metrics to analyze business processes can be classified into four distinct categories: *coupling*, *cohesion*, *complexity*, *modularity* and *size*. In this paper we focus our attention on developing quality metrics to evaluate the complexity of workflow models [7].

Workflow complexity should not be confused with algorithmic complexity measures (e.g. Big-Oh "O"-Notation), whose aim is to compare the performance of algorithms [7]. Workflow complexity can be defined as the degree to which a workflow is difficult to analyze, understand or explain. It can be characterized by the number and intricacy of task interfaces, transitions, conditional and parallel branches, the existence of loops, roles, task categories, the types of data structures, and other workflow characteristics.

In this paper, we present a metric to calculate the Log-Based Complexity (LBC) of workflow patterns [8]. Since our analysis of complexity is based on flow descriptions, we devise complexity metrics for each workflow pattern. The idea of this metric is to relate complexity with the number of different log traces that can be generated from the execution of a workflow. If a workflow always generates the same entries (i.e., the same task ID) in the process log then its complexity is minimal. On the other hand, if a workflow can generate $n!$ distinct log entries (where n is the number of tasks of a workflow) then its complexity is higher.

This paper is structured as follows. The second section presents the related work. In section 3, a new complexity metric for workflow patterns is presented. We start giving a brief overview of what workflow patterns are and explain the reasons why four patterns have not been included in our metric. In section 4, we give a practical example showing how the metric presented is to be applied to workflows. Finally, the last section presents our conclusions.

2 Related work

The concept of process metrics has first been introduced in [7] to provide a quantitative basis for the design, development, validation, and analysis of business process models. Later the concept has been re-coined to Business Process Quality Metrics (BPQM).

The first metric presented in literature was the control-flow complexity (CFC) metric [7]. It was inspired by McCabe's cyclomatic complexity. The CFC metric was evaluated according to Weyuker's properties and an empirical study has been carried out by means of a controlled experiment [9] to validate it. In [10], Mendling proposes a density metric inspired by social network analysis in order to quantify the complexity of an EPC. In [11], the author presents a data flow complexity metric for process models. Reijers and Vanderfeesten [12] also present a metric that computes the degree of coupling and cohesion in a BOM (Bill of materials) model by analyzing data elements. Gruhn and Laue [5] use the notion of cognitive weights as a basic control structure to measure the difficulty in understanding control structures in workflows. Finally, in [6], the authors show how the ProM framework implements some of the quality metrics that have been developed so far.

3 Log-based complexity of workflow patterns

Today, many enterprise information systems store relevant events in a log. The importance of event logs makes them of value and interest to study and to evaluate the complexity of the workflows that generates them. The main idea is to compute the number of distinct logs a specific workflow can generate. The higher the number of distinct logs that can be generated, the more complex the workflow is.

To have an idea on the distinct process logs that can be generated from the execution of a workflow, let us consider the following two examples. A sequential workflow with tasks *A*, *B*, *C*, and *D* can only generate one type of process log entry. Fro example, *A12-B32-C37-D67*. But, if the workflow model defines two sequences: 1) *A* and *B*, and 2) *C* and *D*, and places these two sequences in parallel then the number of different process log entries that can be generated is 6. For example, the entries *A23-B34-C45-D56*, *A23-C45-B34-D56*, *A23-C45-D56-B34*, *C45-D-A23-B34*, *C45-A23-D56-B34*, and *C45-A23-B34-D56*. Intuitively, the second workflow is more complex from a process log perspective since it can have more “mutations”. The first workflow, in our example, is predictable, while the second workflow is unpredictable. As more distinct process log entries can be generated from a workflow, the more unpredictable the workflow is considered to be.

3.1 Workflow patterns

Aalst et al. [13] have identified a number of workflow patterns that describe the behavior of business processes and identify comprehensive workflow functionality. The advantage of these patterns lies in the ability for an in-depth comparison of a number of commercially available workflow management systems based on their capability of executing different workflow structures and their behavior. As we have discussed previously, the log-based complexity is a particular type of control-flow complexity which is influenced by elements such as splits, joins, and loops. Therefore, our first task was to identify the relevant workflow patterns for log-based complexity analysis. We concluded that all patterns, except four, were relevant for the metric we proposed to develop. The Implicit Termination, Multiple Instances without Synchronization, and Cancellation Patterns were not captured by our metric since they are implemented by a very few number of WfMS, the support can lead to an unexpected behavior, or they no not affect the log-based complexity of processes.

3.2 Log-based complexity metrics for workflow patterns

Since it is a well known language, we have used BPMN (Business Process Modeling Notation) to illustrate the log-based complexity of workflow patterns. Of course, we could have used other languages, such as XPDL (XML Process Definition Language), or we could have taken a more formal approach using Petri nets. But we consider that BPMN is a simple and easy language to understand which facilitates readers to comprehend the number of traces introduced by a workflow pattern. To make this paper concise, we will only address a sub-set of workflow patterns. These patterns are representative and explain the rational of our approach to develop the LBC metric.

The simplest element that can generate a log entry is the execution of a task (i.e. an activity). Figure 1 illustrates the representation of a task in BPMN. Please note that the dashed line is not part of the BPMN. We use it to specify the scope of the workflow. In Figure 1, the dashed line specifies that workflow *wf* is composed of task A.

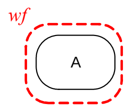

Fig. 1. A task

Since an activity only generates one entry in the process log, its log-based complexity is simply 1, i.e.:

$$LBC_T(wf) = 1$$

Sequence pattern (P1). The sequence pattern is defined as being an ordered series of tasks, with one task starting after a previous task has completed (Figure 2). Please not that BPMN graphically define a sub-workflow using a rounded box with the plus sign (+) inside.

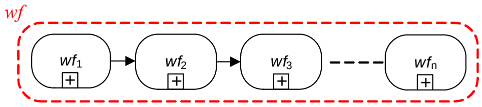

Fig. 2. The sequence pattern

The behavior of this pattern can be described by the use of a token that travels down a sequence from sub-workflow wf_1, to sub-workflow wf_2... and finally reaches sub-workflow wf_n. Since the execution of this pattern always generates the same trace in the process log, the log-based complexity of this pattern is simply given by the following formulae:

$$LBC_{P1}(wf) = \prod_{i=1}^{n} LBC_{x_i}(wf_i)$$

For example, a sequential workflow wf with two sub-workflows wf_1 and wf_2, where wf_1 can generate 4 different traces and wf_2 can generate 3 different traces has a complexity of $LBC(wf) = 4{\cdot}3 = 12$.

Exclusive Choice and Deferred Choice (P4, P16). The exclusive choice pattern (P4, XOR-split) is defined as being a location in the workflow where the flow is split into two or more exclusive alternative paths and, based on a certain condition, one of the paths is taken (Figure 3). The pattern is exclusive since only one of the alternative paths is taken. The deferred choice pattern (P16, a XOR-split abstraction) is very similar to the exclusive choice pattern. In contrast to the exclusive choice pattern, the deferred choice transition selection is based on external input while the exclusive choice relies on information being part of the workflow. Once a transition is activated, the other alternative transitions are deactivated. The moment of choice is delayed until the processing in one of the alternative transitions has actually started.

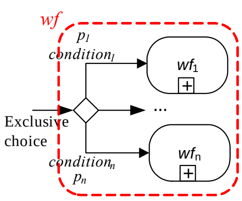

Fig. 3. The exclusive choice pattern

The behavior of these patterns can be described by the use of a token that follows only one of the outgoing transitions of the exclusive choice pattern. Since only one path of the n paths present can be followed, the log-based complexity is the sum of the individual complexity of each workflow $wf_1 \ldots wf_n$. Thus, the LBC for these two patterns is:

$$LBC_{P4}(wf) = LBC_{P16}(wf) = \sum_{i=1}^{n} p_i \times LBC_{x_i}(wf_i)$$

Since workflows are non-deterministic, LBC_{P4} and LBC_{P16} are weighted functions, where p_i is the probability of following a specific path at runtime.

Arbitrary Cycles Loop pattern (P10). The arbitrary cycle pattern is a mechanism for allowing sections of a workflow where one or more activities can be done repeatedly (i.e. a loop). Figure 4 shows an example of the use of the arbitrary cycle pattern.

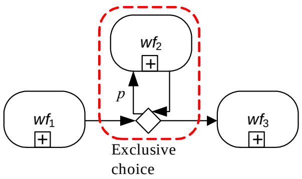

Fig. 4. The arbitrary cycle pattern

At runtime, one of the following scenarios can occur:

wf_1-wf_3	$P_0=1-p$
wf_1-wf_2-wf_3	$P_1=p(1-p) * 1 * LBC_x(wf_2)$
wf_1-wf_2-wf_2-wf_3	$P_2=p^2(1-p) * 2 * LBC_x(wf_2)$
wf_1-wf_2-wf_2-wf_2-wf_3	$P_3=p^3(1-p) * 3 * LBC_x(wf_2)$
…	
wf_1-wf_2-…-wf_2-wf_3	$P_{L-1}=p^{L-1}(1-p) * (L-1) * LBC_x(wf_2)$
wf_1-wf_2-…-wf_2-wf_3	$P_L=(p^L(1-p)+p^{L+1}) * L * LBC_x(wf_2)$

The variable P_j (for $0 \leq j \leq L$, L=maximum number of iterations) indicates the probability of a specific case to occur at runtime when the probabilities of repeating and escaping the loop are p and (1-p), respectively, in every iteration ($0<p<1$). It is assumed to force a compulsorily escape from the loop after L iterations (the probability of such a case is p^{L+1}). Therefore, we can calculate the log-based complexity of the loop as follows:

$$LBC_{P10}(wf) = \left(\sum_{j=0}^{L-1} p^j (1-p) \times j \times LBC_x(wf_2) \right) + (p^L(1-p) + p^{L+1}) \times L \times LBC_x(wf_2)$$

Interleaved parallel routing pattern (P17). In this pattern, a set of activities is executed with no specific order. The performers of the activities will decide the order of the activities. Each task in the set is executed and no two activities are executed at the same moment. It is not until one task is completed that the decision on what to do next is taken.

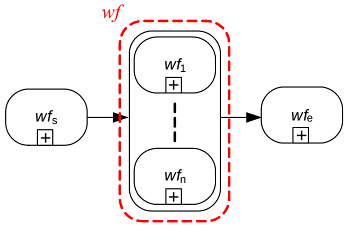

Fig. 5. The interleaved parallel routing pattern

Figure 5 illustrates the interleaved parallel routing pattern. Once sub-workflow wf_s is completed, a token is transferred to the set of sub-workflow wf_1, …, wf_n. The token will be assigned to one of the sub-worklfows wf_1, wf_2,…, or wf_n and then transferred to another sub-workflow until all the sub-workflows are completed. This is done sequentially. Since all sub-workflows will be activated at some point in time in any order, we have n! permutations for the sub-workflows, therefore the log-based complexity is:

$$LBC_{P17}(wf) = n! \times \prod_{i=1}^{n} LBC_{x_i}(wf_i)$$

4 Aggregating the complexity of workflow patterns

Having devised custom metrics for each workflow pattern, we can calculate the LBC of workflows. Our approach to calculate the overall log-based complexity of a workflow consists in the stepwise collapsing of the workflow into a single node by alternately aggregating workflow patterns. The algorithm that we use repeatedly applies a set of workflow transformation rules (based on the workflow patterns that we have analyzed) to a workflow until only one atomic task remains. Each time a transformation rule is applied, the workflow structure changes. After several iterations only one task will remain. When this state is reached, the remaining task contains the complexity corresponding to the initial workflow under analysis.

Figure 6 illustrates the set of transformation rules that are applied to an initial workflow to compute the log-based complexity. To the initial process, illustrated in Figure 12.a), we apply patterns LBC_T and LBC_{P13}. The resulting process is illustrated in Figure 12.b). To this new process we apply patterns LBC_T, LBC_{P1}, LBC_{P5}, and LBC_{P13}. The process suffers various transformations as shown in Figures 12.c) and Figure 12.d). Finally, after the last transformation, only one task remains (Figure 12.e) and this task (ABCDEnEF) contains the overall complexity of the workflow which is 5.75. This indicates that the initial workflow can generates, on average (since the workflow is non-deterministic) 5.75 distinct process logs.

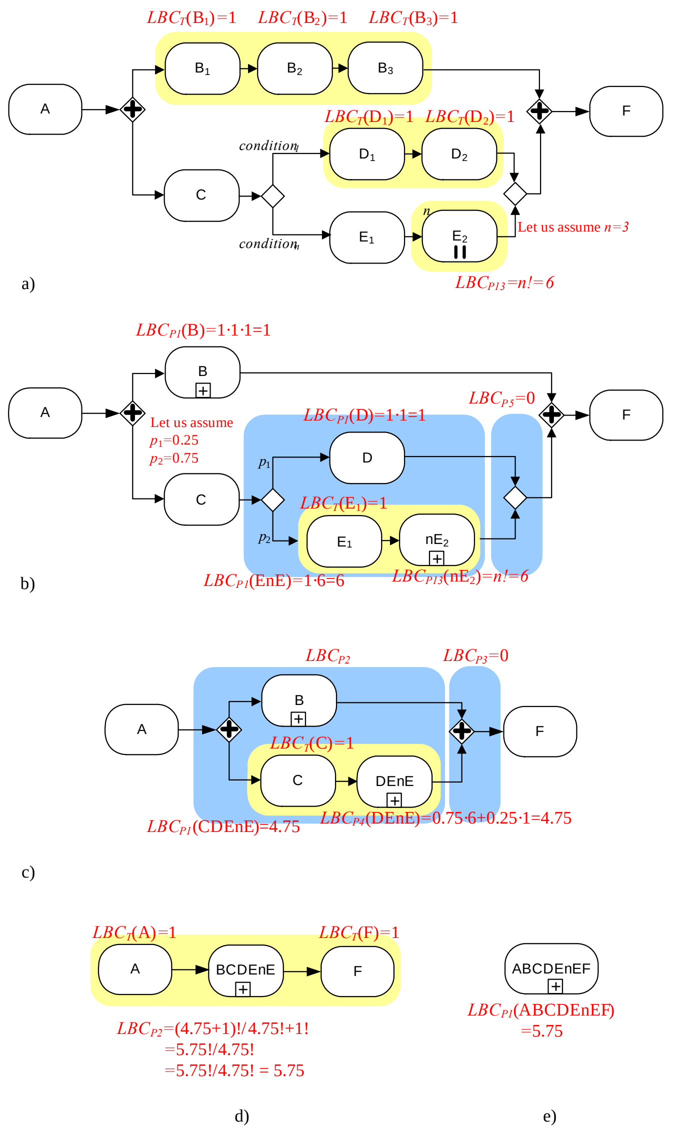

Fig. 6. Log-based complexity computation

5 Conclusions

Recently, a new approach to workflow analysis has been proposed and targets the development of Business Process Quality Metrics (BPQM) to evaluate workflow models. One particular class of quality metrics has the goal of analyzing the complexity of workflow models. This analysis enables to identify complex workflows that require reparative actions to improve their comprehensibility. To enlarge the number of approaches available to analyze workflows, in this paper, we presented the log-based complexity (LBC) metric to calculate the complexity of workflows. Our approach consisted of devising a complexity metric based on the number of process logs that are generated when workflows are executed. Our complexity metric is a design-time measurement and can be used to evaluate the difficulty of producing a workflow design before its implementation.

This work was partially funded by FCT, POCTI-219, and FEDER.

References

1. Verbeek, H.M.W., T. Basten, and W.M.V.d. Aalst, Diagnosing workflow processes using woflan. The Computer Journal, 2001. 44(4): p. 246-279.
2. Gruhn, V. and R. Laue. Adopting the Cognitive Complexity Measure for Business Process Models. in 5th IEEE International Conference on Cognitive Informatics. 2006. Beijing, China: IEEE Computer Society
3. Latva-Koivisto, A.M., Finding a complexity measure for business process models. 2001, Helsinki University of Technology, Systems Analysis Laboratory: Helsinki
4. Cardoso, J., et al. A Discourse on Complexity of Process Models. in BPI'06 - Second International Workshop on Business Process Intelligence, In conjunction with BPM 2006. 2006. Vienna, Austria: Springer-Verlag, Berlin, Heidelberg.
5. Gruhn, V. and R. Laue. Complexity Metrics for Business Process Models. in 9th International Conference on Business Information Systems. 2006. Klagenfurt, Austria: GI.
6. Vanderfeesten, I., et al., Quality Metrics for Business Process Models, in Workflow Handbook 2007, L. Fischer, Editor. 2007, Future Strategies Inc.: Lighthouse Point, FL, USA. p. 179-190.
7. Cardoso, J., Evaluating Workflows and Web Process Complexity, in Workflow Handbook 2005, L. Fischer, Editor. 2005, Future Strategies Inc.: Lighthouse Point, FL, USA. p. 284-290.
8. Aalst, W.M.P.v.d., et al., Workflow Patterns. Distributed and Parallel Databases, 2003. 14(3): p. 5-51.
9. Cardoso, J. Process control-flow complexity metric: An empirical validation. in IEEE International Conference on Services Computing (IEEE SCC 06). 2006. Chicago, USA: IEEE Computer Society.
10. Mendling, J., Testing Density as a Complexity Metric for EPCs, Technical Report JM-2006-11-15. 2006, Vienna University of Economics and Business Administration, Austria.
11. Cardoso, J. About the Data-Flow Complexity of Web Processes. in 6th International Workshop on Business Process Modeling, Development, and Support: Business Processes and Support Systems: Design for Flexibility. 2005. Porto, Portugal.
12. Reijers, H.A. and I.T.P. Vanderfeesten, Cohesion and Coupling Metrics for Workflow Process Design, in BPM 2004 (LNCS 3080), J. Desel, B. Pernici, and M. Weske, Editors. 2004, Springer-Verlag: Berlin, Heidelberg. p. 290-305.
13. Aalst, W.M.P.v.d., et al. Advanced Workflow Patterns. in Seventh IFCIS International Conference on Cooperative Information Systems. 2000. Eilat, Israel.