

Workflow Quality of Service Management using Data Mining Techniques

Jorge Cardoso

Abstract— Organizations have been aware of the importance of Quality of Service (QoS) for competitiveness for some time. It has been widely recognized that workflow systems are a suitable solution for managing the QoS of processes and workflows. The correct management of the QoS of workflows allows for organizations to increase customer satisfaction, reduce internal costs, and increase added value services. In this paper we show a novel method, composed of several phases, describing how organizations can apply data mining algorithms to predict the QoS for their running workflow instances. Our method has been validated using experimentation by applying different data mining algorithms to predict the QoS of workflow.

Index Terms—Quality of Service, Data Mining, Business Process, Workflow.

I. INTRODUCTION

The increasingly global economy requires advanced information systems. Business Process Management Systems (BPMS) provide a fundamental infrastructure to define and manage several types of business processes. BPMS, such as Workflow Management Systems (WfMS), have become a serious competitive factor for many organizations that are increasingly faced with the challenge of managing e-business applications, workflows, Web services, and Web processes. WfMS allow organizations to streamline and automate business processes and reengineer their structure; in addition, they increase efficiency and reduce costs.

One important requirement for BPMS and WfMS is the ability to manage the Quality of Service (QoS) of processes and workflows [1]. The design and composition of processes cannot be undertaken while ignoring the importance of QoS measurements. Appropriate control of quality leads to the creation of quality products and services; these, in turn, fulfill customer expectations and achieve customer satisfaction. It is not sufficient to just describe the logical or operational functionality of activities and workflows. Rather, design of workflows must include QoS specifications, such as response time, reliability, cost, and so forth. In our work, QoS covers costs and time.

One important activity, under the umbrella of QoS management, is the prediction of the QoS of workflows. Several approaches can be identified to predict the QoS of workflows before they are invoked or during their execution, including statistical algorithms [1], simulation

[2], and data mining based methods [3, 4].

The latter approach, which uses data mining methods to predict the QoS of workflows, has received significant attention and has been associated with a recent new area coined as Business Process Intelligence (BPI). In this paper, we investigate the enhancements that can be made to previous work on BPI and business process quality to develop more accurate prediction methods. The methods presented in [3, 4] can be extended and refined to provide a more flexible approach to predict the QoS of workflows.

This paper is structured as follows: In Section 2, we present the motivation for our work. In Section 3, we explain our method of carrying out QoS mining based on path mining, QoS activity models, and workflow QoS estimation. Section 4 briefly indicates the results of our experiments and presents the related work in this area. Finally, section 5 presents our conclusions.

II. MOTIVATION

Nowadays, a considerable number of organizations are adopting workflow management systems to support their business processes. The current systems available manage the execution of workflow instances without any quality of service management on important parameters such as delivery deadlines, reliability, and cost of service.

Let us assume that a workflow is started to deliver a particular service to a customer. It would be helpful for the organization supplying the service to be able to predict how long the workflow instance will take to be completed or the cost associated with its execution. Since workflows are non-deterministic and concurrent, the time it takes for a workflow to be completed and its cost depends not only on which activities are invoked during the execution of the workflow instance, but also depends on the time/cost of its activities. Predicting the QoS that a workflow instance will exhibit at runtime is a challenge because a workflow schema w can be used to generate n instances, and several instances w_i ($i \leq n$) can invoke a different subset of activities from w . Therefore, even if the time and cost associated with the execution of activities were static, the QoS of the execution of a workflow would vary depending on the activities invoked at runtime.

For organizations, being able to predict the QoS of workflows has several advantages. For example, it is possible to monitor and predict the QoS of workflows at any time. Workflows must be rigorously and constantly monitored throughout their life cycles to assure compliance both with initial QoS requirements and targeted objectives. If a workflow management system identifies that a running workflow will not meet initial QoS requirements, then

J. Cardoso is with the Department of Mathematics and Engineering, University of Madeira, 9050-390 Funchal, Portugal (e-mail: jcardoso@uma.pt).

adaptation strategies [5] need to be triggered to change the structure of a workflow instance. By changing the structure of a workflow we can reduce its cost or execution time.

III. QoS MINING

In this section we focus on describing a new method that can be used by organizations to apply data mining algorithms to historical data and predict QoS for their running workflow instances. The method presented in this paper constitutes a major and significant difference from the method described in [4]. The method is composed of three distinct phases (figure 1) that will be explained in the following subsections.

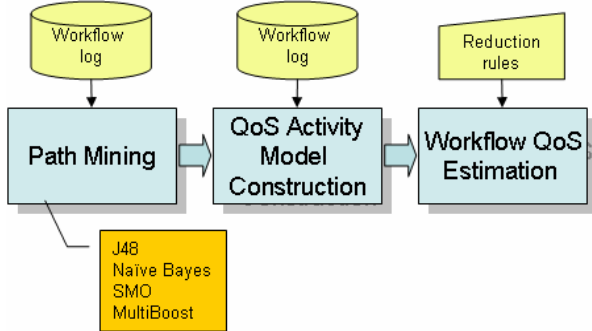


Fig. 1. Phases of workflow QoS mining

In the first phase, the workflow log is analyzed and data mining algorithms are applied to predict the path that will be followed by workflow instances at runtime. This is called path mining. Path mining identifies which activities will most likely be executed in the context of a workflow instance. Once we know the path, we also know the activities that will be invoked at runtime. For each activity we construct a QoS activity model based on historical data (also known as temporal or transactional) which describes the runtime behavior (duration and cost) of an activity. In the last phase, we compute the QoS of the overall workflow based on the path predicted and from the QoS activity models using a set of reduction rules.

A. Path Mining

As we have stated previously, the QoS of a workflow is directly dependent on which activities are invoked during its execution. Different sets of activities can be invoked at runtime because workflows are non-deterministic. Path mining [6, 7] uses data mining algorithms to predict which path will be followed when executing a workflow instance.

A path is composed of a set of activities invoked and executed at runtime by a workflow. For example, when path mining is applied to the simple workflow illustrated in figure 2, the workflow management system can predict the probability of paths A, B, and C being followed at runtime. Paths A and B have each 6 activities, while path C has only 4 activities. In figure 2, the symbol \oplus represented non-determinism (i.e., a xor-split or xor-join).

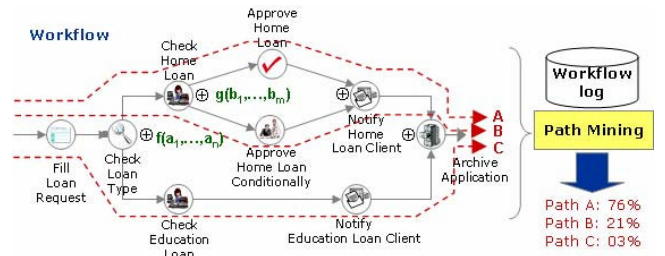


Fig. 2. Path mining

To perform path mining, current workflow logs need to be extended to store information indicating the values and the type of the input parameters passed to activities and the output parameters received from activities. The values of inputs/outputs are generated at runtime during the execution of workflow instances. Table 1 shows an extended workflow log which accommodates input/output values of activity parameters that have been generated at runtime. Each ‘Parameter/Value’ entry as a type, a parameter name, and a value (for example, string loan-type=“car-loan”).

TABLE I
EXTENDED WORKFLOW LOG
Workflow
log extension

...	Parameter/Value	Path
...	int SSN=7774443333; string loan-type="car-loan"
...	string name=jf@uma.pt; ...	{FillLoanRequest, CheckLoanType, CheckCarLoan, ApproveCarLoan, NotifyCarLoanClient, ArchiveApplication}
...

Additionally, the log needs to include path information: a path describing the activities that have been executed during the enactment of a process. This information can easily be stored in the log. From the implementation perspective it is space efficient to store in the log only the relative path, relative to the previous activity, not the full path. Table 1 shows the full path approach because it is easier to understand how paths are stored in the log.

During this phase, and compared to [3, 4], we only need to add information on paths to the log. Once enough data is gathered in the workflow log, we can apply data mining methods to predict the path followed by a process instance at runtime based on instance parameters. In section 4.2, we will show how the extended workflow log can be transformed to a set of data mining instances. Each data mining instance will constitute the input to machine learning algorithm.

B. QoS activity model construction

After carrying out path mining, we know which activities a workflow instance will be invoking in the near future. For each activity that will potentially be invoked we build what we call a QoS activity model. The model includes information about the activity behavior at runtime, such as its cost and the time the activity will take to execute [1].

Each QoS activity model can be constructed by carrying

out activity profiling. This technique is similar to the one used to construct operational profiles. Operational profiles have been proposed by Musa [8, 9] to accurately predict future the reliability of applications. The idea is to test the activity based on specific inputs. In an operational profile, the input space is partitioned into domains, and each input is associated with a probability of being selected during operational use. The probability is employed in the input domain to guide input generation. The density function built from the probabilities is called the operational profile of the activity. At runtime, activities have a probability associated with each input. Musa [9] described a detailed procedure for developing a practical operational profile for testing purposes. In our case, we are interested in predicting, not the reliability, but the cost and time associated with the execution of workflow activities.

During the graphical design of a workflow, the business analyst and domain expert construct a QoS activity model for each activity using activity profiles and empirical knowledge about activities. The construction of a QoS model for activities is made at design time and re-computed at runtime, when activities are executed. Since the initial QoS estimates may not remain valid over time, the QoS of activities is periodically re-computed, based on the data of previous instance executions stored in the workflow log.

The re-computation of QoS activity metrics is based on data coming from designer specifications (i.e. the initial QoS activity model) and from the workflow log. Depending on the workflow data available, four scenarios can occur (Table II): a) For a specific activity a and a particular dimension Dim (i.e., time or cost), the average is calculated based only on information introduced by the designer (Designer Average $_{Dim}(a)$); b) the average of an activity a dimension is calculated based on all its executions independently of the workflow that executed it (Multi-Workflow Average $_{Dim}(a)$); c) the average of the dimension Dim is calculated based on all the times activity a was executed in any instance from workflow w (Workflow Average $_{Dim}(t, w)$); and d) the average of the dimension of all the times activity t was executed in instance i of workflow w (Instance Average $_{Dim}(t, w, i)$).

TABLE II
QoS DIMENSIONS COMPUTED AT RUNTIME

a)	$QoS_{Dim}(a) =$	Designer Average $_{Dim}(a)$
b)	$QoS_{Dim}(a) =$	$w_1 * \text{Designer Average}_{Dim}(a) +$ $w_2 * \text{Multi-Workflow Average}_{Dim}(a)$
c)	$QoS_{Dim}(a, w) =$	$w_1 * \text{Designer Average}_{Dim}(a) +$ $w_2 * \text{Multi-Workflow Average}_{Dim}(a) +$ $w_3 * \text{Workflow Average}_{Dim}(a, w)$
d)	$QoS_{Dim}(a, w, i)$ $=$	$w_1 * \text{Designer Average}_{Dim}(a) +$ $w_2 * \text{Multi-Workflow Average}_{Dim}(a) +$ $w_3 * \text{Workflow Average}_{Dim}(a, w) +$ $w_4 * \text{Instance Workflow Average}_{Dim}(a, w, i)$

Let us assume that we have an instance i of workflow w running and that we desire to predict the QoS of activity $a \in w$. The following rules are used to choose which formula to apply when predicting QoS. If activity a has never been executed before, then formula a) is chosen to predict activity QoS, since there is no other data available in the workflow log. If activity a has been executed previously, but in the context of workflow w_n , and $w \neq w_n$, then

formula b) is chosen. In this case we can assume that the execution of a in workflow w_n will give a good indication of its behavior in workflow w . If activity a has been previously executed in the context of workflow w , but not from instance i , then formula c) is chosen. Finally, if activity a has been previously executed in the context of workflow w , and instance i , meaning that a loop has been executed, then formula d) is used.

The workflow management system uses the formulae from Table II to predict the QoS of activities. The weights w_k are manually set. They reflect the degree of correlation between the workflow under analysis and other workflows for which a set of common activities is shared. At this end of this second phase, we already know the activities of a workflow instance that will most likely be executed at runtime, and for each activity we have a model of its QoS, i.e. we know the time and cost associated with the invocation of the activity.

C. Workflow QoS Estimation

Once we know the path, i.e. the set of activities which will be executed by a workflow instance, and we have a QoS activity model for each activity, we have all the elements required to predict the QoS associated with the execution of a workflow instance.

To compute the estimated QoS of a process in execution, we use a variation of the Stochastic Workflow Reduction (SWR) algorithm [1]. The variation of the SWR algorithm that we use does not include probabilistic information about transitions. The SWR is an algorithm for computing aggregate QoS properties step-by-step. At each step a reduction rule is applied to shrink the process. At each step the time and cost of the activities involved is computed. This is continued until only one activity is left in the process. When this state is reached, the remaining activity contains the QoS metrics corresponding to the workflow under analysis. For the reader interested in the behavior of the SWR algorithm we refer to [1].

For example, if the path predicted in the first phase of our QoS mining method includes a parallel system, as show in Figure 3, the parallel system reduction rule is applied to a part of the original workflow (Figure 3.a) and a new section of the workflow is created (Figure 3.b).

A system of parallel activities t_1, t_2, \dots, t_n , an *and* split activity t_a , and an *and* join activity t_b can be reduced to a sequence of three activities t_a, t_{1n} , and t_b . In this reduction, the incoming transitions of t_a and the outgoing transition of activities t_b remain the same. The only outgoing transitions from activity t_a and the only incoming transitions from activity t_b are the ones shown in the figure below.

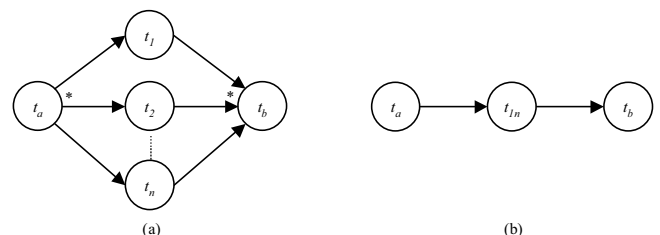


Fig. 3. Parallel system reduction

The QoS of the new workflow is computed using the

following formulae (the QoS of tasks t_a and t_b remain unchanged):

$$\begin{aligned} \text{Time}(t_{in}) &= \text{Max}_{i \in \{1..n\}} \{\text{Time}(t_i)\} \text{ and} \\ \text{Cost}(t_{in}) &= \sum_{1 \leq i \leq n} \text{Cost}(t_i) \end{aligned}$$

Reduction rules exist for sequential, parallel, conditional, loop, and network systems [1]. These systems or pattern are fundamental since a study on fifteen major workflow management systems [10] showed that most systems support the reduction rules presented. Nevertheless, additional reduction rules can be developed to cope with the characteristics and features of specific workflow systems.

Our approach to workflow QoS estimation – which uses a variation of the SWR algorithm – shows that the prediction of workflow QoS can be used to obtain actual metrics (e.g. the workflow instance w will take 3 days and 8 hours to execute) and not only information that indicates if an instance takes “more” than D days or “less” than D days to execute.

VI. EXPERIMENTS AND RELATED WORK

To valid our method, we have carried out experiments using the loan process application described in [6]. The results have clearly show that the QoS mining method yields estimations that are very close to the real QoS of the running processes.

Process and workflow mining is addressed in several papers and a detailed survey of this research area is provided in [11]. In [3, 4], a Business Process Intelligence (BPI) tool suite that uses data mining algorithms to support process execution by providing several features, such as analysis and prediction is presented. In [12] and [13] a machine learning component able to acquire and adapt a workflow model from observations of enacted workflow instances is described. Agrawal, Gunopulos et al. [14] propose an algorithm that allows the user to use existing workflow execution logs to automatically model a given business process presented as a graph. Chandrasekaran et al., [2] describe a simulation coupled with a Web Process Design Tool (WPDT) and a QoS model [1] to automatically simulate and analyze the QoS of Web processes. While the research on QoS for BMPS is limited, the research on time management, which is under the umbrella of QoS process, has been more active and productive. Eder et al. [15] and Pozewaunig et al. [16] present an extension of CMP and PERT frameworks by annotating workflow graphs with time, in order to check the validity of time constraints at process build-time.

V. CONCLUSIONS

The importance of QoS (Quality of Service) management for organizations and for workflow systems has already been much recognized by academia and industry. The design and execution of workflows cannot be undertaken while ignoring the importance of QoS measurements since they directly impact the success of organizations. In this paper we have shown a novel method

that allows us to achieve high levels of accuracy when predicting the QoS of workflows. Our conclusion indicates that workflow QoS mining should not be applied as a one-step methodology to workflow logs. Instead, if we use a methodology that includes path mining, QoS activity models, and workflow QoS estimation, we can obtain a very good prediction accuracy.

REFERENCES

- [1]. Cardoso, J., et al., *Modeling Quality of Service for workflows and web service processes*. Web Semantics: Science, Services and Agents on the World Wide Web Journal, 2004. **1**(3): p. 281-308.
- [2]. Chandrasekaran, S., et al. *Service Technologies and their Synergy with Simulation*. in *Proceedings of the 2002 Winter Simulation Conference (WSC'02)*. 2002. San Diego, California.
- [3]. Grigori, D., et al., *Business Process Intelligence*. Computers in Industry, 2004. **53**: p. 321–343.
- [4]. Grigori, D., et al. *Improving Business Process Quality through Exception Understanding, Prediction, and Prevention*. in *27th VLDB Conference*. 2001. Roma, Italy.
- [5]. Cardoso, J. and A. Sheth. *Adaptation and Workflow Management Systems*. in *International Conference WWW/Internet 2005*. 2005. Lisbon, Portugal.
- [6]. Cardoso, J., *Path Mining in Web processes using Profiles*, in *Encyclopedia of Data Warehousing and Mining*, J. Wang, Editor. 2005, Idea Group Inc. p. 896-901.
- [7]. Cardoso, J. and M. Lenic, *Web Process and Workflow Path mining using the multimethod approach*. Journal of Business Intelligence and Data Mining (JBIDM). submitted., 2005.
- [8]. Musa, J.D., *Operational Profiles in Software-Reliability Engineering*. IEEE Software, 1993. **10**(2): p. 14-32.
- [9]. Musa, J.D., *Software reliability engineering: more reliable software, faster development and testing*. 1999, New York: McGraw-Hill.
- [10]. Aalst, W.M.P.v.d., et al., *Workflow patterns homepage*. 2002, <http://tmitwww.tm.tue.nl/research/patterns>.
- [11]. Aalst, W.M.P.v.d., et al., *Workflow Mining: A Survey of Issues and Approaches*. Data & Knowledge Engineering (Elsevier), 2003. **47**(2): p. 237-267.
- [12]. Herbst, J. and D. Karagiannis. *Integrating Machine Learning and Workflow Management to Support Acquisition and Adaption of Workflow Models*. in *Ninth International Workshop on Database and Expert Systems Applications*. 1998.
- [13]. Weijters, T. and W.M.P. van der Aalst. *Process Mining: Discovering Workflow Models from Event-Based Data*. in *13th Belgium-Netherlands Conference on Artificial Intelligence (BNAIC 2001)*. 2001. Amsterdam, The Netherlands.
- [14]. Agrawal, R., D. Gunopulos, and F. Leymann. *Mining Process Models from Workflow Logs*. in *Sixth International Conference on Extending Database Technology*. 1998. Valencia, Spain: Springer.
- [15]. Eder, J., et al. *Time Management in Workflow Systems*. in *BIS'99 3rd International Conference on Business Information Systems*. 1999. Poznan, Poland: Springer Verlag.
- [16]. Pozewaunig, H., J. Eder, and W. Liebhart. *ePERT: Extending PERT for workflow management systems*. in *First European Symposium in Advances in Databases and Information Systems (ADBIS)*. 1997. St. Petersburg, Russia.