Workflow Quality of Service

Jorge Cardoso, Amit Sheth, and John Miller

LSDIS Lab, University of Georgia, USA, amit@cs.uga.edu

Abstract: Workflow management systems (WfMSs) have been used to support various types of business processes for more than a decade now. In e-commerce processes, suppliers and customers define a binding agreement or contract between the two parties, specifying quality of service (QoS) items such as products or services to be delivered, deadlines, quality of products, and cost of service. Organizations operating in modern markets require an excellent degree of quality of service management. A good management leads to the creation of quality products and services, which in turn fulfills customer expectations and achieves customer satisfaction. Therefore, when services or products are created or managed using workflow processes, the underlying WfMS must accept the specification, be able to predict, monitor, and control the QoS rendered to customers. To achieve these objectives the first step is to develop an adequate QoS model for workflow processes and develop methods to compute QoS.

1 INTRODUCTION

Organizations are constantly seeking new and innovative information systems to better fulfill their mission and strategic goals. In the past decade, Workflow Management Systems (WfMSs) have been distinguished due to their significance and impact on organizations. WfMSs allow organizations to streamline and automate business processes, reengineer their structure, as well as, increase efficiency and reduce costs.

Our experience with real world enactment services (Miller, et al, 1998, Kochut, et al, 1999) and applications made us aware of the importance of Quality of Service (QoS) management for workflow systems. For organizations, being able to characterize workflows based on their QoS has three direct advantages. First, it allows organizations to translate more efficiently

The original version of this chapter was revised: The copyright line was incorrect. This has been corrected. The Erratum to this chapter is available at DOI: [10.1007/978-0-387-35621-1_43](https://doi.org/10.1007/978-0-387-35621-1_43)

their vision into their business processes, since workflow can be designed according to QoS metrics. Second, it allows the selection and execution of workflows based on their QoS to better fulfill customers expectations. Third, it also makes possible the monitoring of workflows based on QoS, allowing managers to set compensation strategies when undesired metrics are identified.

Our goal is to develop a workflow QoS specification and methods to predict, analyze and monitor QoS. We start by investigating the relevant quality of service dimensions, which are necessary to correctly characterize workflows. Once the QoS dimensions are identified, it is necessary to devise methodologies to estimate QoS for tasks. Finally, algorithms and methods need to be developed to compute workflow QoS. In workflows, quality metrics estimates are associated with tasks and tasks compose workflows. The computation of workflow QoS is done based on the QoS of the tasks that compose a workflow.

This paper is structured as follows. Section 2 introduces our workflow QoS model and describes each of its dimensions. In section 3, we describe how QoS estimates for tasks are created. Section 4 discusses two techniques to compute workflow QoS from task QoS. Section 5 discusses related work in this area and section 6 presents our conclusions.

2 WORKFLOW QUALITY OF SERVICE

For us, workflow QoS represents *the quantitative and qualitative characteristics of a workflow application necessary to achieve a set of initial requirements*. Workflow QoS addresses the non-functional issues of workflows, rather than workflow process operations. Quantitative characteristics can be evaluated in terms of concrete measures such as workflow execution time, cost, etc. Qualitative characteristics specify the expected services offered by the system such as security and fault-tolerance mechanisms.

Workflow QoS is composed of different dimensions that are used to characterize workflow schema and instances. To our knowledge most of the research carried out to extend workflow systems capabilities, in the context of QoS, has only been done for the time dimension (Bussler, 1998, Dadam, et al 2000, Eder, et al, 1999, Kao, Garcia Molina, 1993, Marjanovic, et al 1999, Sadig, et al, 2000, Son, et al, 2001), which is only one of the dimensions under the workflow QoS umbrella. Even though some WfMSs currently offer time management support, the technology available is rudimentary (Eder, et al, 1999). The Crossflow project (Damen, et al, 2000, Grefen, et al, 2000, Klingemann, et al, 1999) is the one that most closely relates to our work, for which workflow cost is also considered.

Quality of service can be characterized along to various dimensions. We have investigated related work to decide which dimensions would be relevant to compose our QoS model. Our research targeted two distinct areas: operation management for organizations and quality of service for software systems (which include networking, middleware areas, and real-time applications.) The study of those two areas is important, since workflow systems are widely used to model organizational business processes, and workflow systems are themselves software systems.

2.1 QoS Model

Based on previous studies and our experience in the workflow domain, we construct a QoS model composed of four dimensions: *time*, *cost*, *fidelity*, and *reliability*.

Time (T) is a common and universal measure of performance. For workflow systems, it can be defined as the total time needed by an instance to transform a set of inputs into outputs. Task response time (T) corresponds to the time an instance spends to be processed by a task. The *task response time* can be broken down into two major components: *delay time* and *process time*. Delay time (DT) refer to non-value-add time needed in order for an instance to be processed by a task. While, those two metrics are part of the task operation, they do not add any value to it. Process time (PT) is the time a workflow instance spends at a task while being processed, in other words it corresponds to the time a task needs to process an instance.

Cost (C) represents the cost associated with the execution of workflow tasks. During workflow design, prior to workflow instantiation and during workflow execution it is necessary to estimate the cost of its execution to guarantee that financial plans are followed. Task cost is the cost incurred when a task t is executed, and can be broken down into two major components: *enactment cost* and *realization cost*. The enactment cost (EC) is the cost associated with the management of the workflow system and workflow instances monitoring. The realization cost (RC) is the cost associated with the runtime execution of the task.

We view **Fidelity (F)** as a function of effective design and refer to an intrinsic property or characteristic of a good produced or service rendered. Fidelity is often difficult to define and measure because it is subjective to judgments and perceptions. Nevertheless, the fidelity of workflows must be predicted, when possible, and carefully controlled when needed. Workflow tasks have a fidelity vector dimension composed by a set of fidelity attributes $(F(t).a_i)$, to reflect and quantify task operations. Each fidelity attribute refers to a property or characteristic of the product being created, transformed, or analyzed. Fidelity attributes are used by the workflow system to

compute how well workflows, instances, and tasks are meeting user specifications.

Task **Reliability** (R) corresponds to the likelihood that the components will perform when the user demands it and it is a function of the failure rate. This QoS dimension provides information concerning the relationship between the number of times a task reaches the state done or committed, and the number of times it reaches the failed/aborted state.

2.2 QoS and Production Workflows

One of the most popular workflow classifications distinguishes between *ad hoc* workflows, administrative workflows, and production workflows. This classification was first mentioned by (McCready, 1992). The main differences between these types include structure, repetitiveness, predictability, complexity, and degree of automation.

The QoS model presented is better suited for production workflows (McCready, 1992) since they are more structured, predictable, and repetitive. Production workflows involve complex and highly structured processes, whose execution requires a high number of transaction accessing different information systems. These characteristics allow the construction of adequate QoS models for workflow tasks. In the case of *ad hoc* workflows, the information, the behavior, and the timing of tasks are largely unstructured, which makes the procedure of constructing a good QoS model more difficult and complex.

3 CREATION OF QOS ESTIMATES

Determining useful estimates for the QoS properties of a task can be challenging. A combination of *a priori* estimates from designers as well as estimates computed from prior executions will be used, with the historical data playing a larger role as more data is collected. Additional complexities are due to the fact that QoS is parametric. For example, the response time of a service that takes an XML document as input will depend on the size of the document. Estimates for workflows can be developed in two ways: (a) estimates for the entire workflow can be created just like they are for ordinary/atomic services (*i.e.*, *a priori* estimates refined as execution monitoring data is collected), (b) the QoS properties can be synthesized from the QoS properties of the tasks making up the workflow. Synthesizing aggregate estimates requires several problems to be solved, among them (1) determination of transitions probabilities from transitions conditions and (2) dealing with correlation between individual tasks.

In order to facilitate the analysis of workflow QoS, it is necessary to initialize task QoS metrics and also initialize stochastic information indicating the probability of transitions being fired at runtime. Once tasks and transitions have their estimates set, algorithms and mechanisms such as simulation can be applied to compute overall workflow QoS.

3.1 QoS for Tasks

Task QoS is initialized at design time and re-computed at runtime when tasks are executed. During the graphical construction of a workflow process, each task receives information estimating its quality of service behavior at runtime. The re-computation of QoS task metrics is based on data coming from the user specifications and from the workflow system log.

3.2 QoS for Transitions

The same methodology used to estimate task QoS, is also used to estimate workflow transitions probabilities. The user initializes the transitions probabilities at design time. At runtime the probabilities are re-computed. If a workflow has never been executed, the values for the transitions are obviously taken from initial user specifications. If instances of a workflow w have already been executed, then the data used to re-compute the probabilities comes from initial user specifications for workflow w and from completed instances.

4 QOS COMPUTATION

Once QoS estimates for tasks and for transitions are determined we can compute overall workflow QoS. We describe two methods that can be used to compute QoS metrics for a given workflow process: analysis and simulation. The selection of one of the methods is based on a tradeoff between time and accuracy of results. The analytic method is computationally faster, but yields results, which may not be as accurate as the ones obtained with simulation.

4.1 Analytic Models

Comprehensive solutions to the difficult problems encountered in synthesizing QoS for composite services are discussed in detail (Cardoso, et al, 2002). This work presents a stochastic workflow reduction algorithm (SWR)

(Cardoso, 2002) for computing aggregate QoS properties step-by-step. At each step a reduction rule is applied to shrink the network. At each step the response time (RT), processing time (PT), delay time (DT), cost (C) and reliability (R) of the tasks involved is computed. Additional task metrics can also be computed, such as task enactment time and setup time. This is continued until only one atomic task (Kochut, et al, 1999) is left in the network. When this state is reached, the remaining task contains the QoS metrics corresponding to the workflow under analysis. The set of reduction rules that can be applied to a composite service (network) corresponds to the set of inverse operation that can be used to construct a network of services. We have decided to only allow the construction of workflows based on a set of predefined construction rules to protect users from designing invalid workflows. The algorithm uses a set of six distinct reduction rules: (1) sequential, (2) parallel, (3) conditional, (4) fault-tolerant, (5) loop, and (6) network.

4.2 Simulation Models

While analytical methods can be effectively used, another alternative is to utilize simulation analysis (Miller, et al, 2002). Simulation can play an important role in tuning quality of service metrics of workflows by exploring “what-if” questions. When the need to adapt or to change a workflow is detected, deciding what changes to carry out can be very difficult. Before a change is actually made, its possible effects can be explored with simulation. To facilitate rapid feedback, the workflow system and simulation system need to interoperate. In particular, workflow specification documents need to be translated into simulation model specification documents so that the new model can be executed/animated on the fly.

In our project, these capabilities involve a loosely coupled integration between the WfMS (METEOR, Miller, et al, 1998, Kochut, et al, 1999) and the simulation system (JSIM, (Miller, et al, 2002)). Workflow is concerned with scheduling and transformations that take place in tasks, while simulation is mainly concerned with system performance. For modeling purposes, a workflow can be abstractly represented by using directed graphs (*e.g.*, one for control flow and one for data flow, or one for both). Since both models are represented as directed graphs interoperation is facilitated. In order to carry out a simulation, the appropriate workflow model is retrieved from the repository and translated into a JSIM simulation model specification. The simulation model is displayed graphically and then executed/animated. Statistical results are collected and displayed which indicate workflows QoS.

5 RELATED WORK

The work found in the literature on quality of service for WfMS is limited. The Crossflow project (Damen, et al, 2000, Grefen, et al, 2000, Klingemann, et al, 1999) has given a major contribution. In their approach, a continuous-time Markov chain (CTMC) is used to calculate the time and cost associated with workflow executions. While the research on quality of service for WfMS is limited, the research on time management, which is under the umbrella of workflow QoS, has been more active and productive. Eder (1999) and Pozewaunig (1997) present an extension to CMP and PERT by annotating workflow graphs with time. At process build-time, instantiation-time, and runtime the annotations are used to check the validity of time constraints. The major limitation of their approach is that only direct acyclic graphs (DAG) can be modeled. This is a significant limitation since the majority of workflows have cyclic graphs. Cycles are in general used to represent rework actions or repetitive activities within a workflow. Marjanovic, Orlowska, (1999) describe a workflow model enriched with modeling constructs and algorithms for checking the consistency of workflow temporal constraints. Son, (2001) present a solution for the deadline allocation problem based on queuing networks.

Recently, in the area of Web services, researchers have also manifested an interest for QoS. The DAML-S, (2001) specification allows the semantic description of business processes. The specification includes constructs to specify quality of service parameters, such as quality guarantees, quality rating, and degree of quality. While DAML-S has identified specification for Web service and business processes as a key specification component, the QoS model adopted should be significantly improved to supply a realistic solution to its users. One current limitation of DAML-S' QoS model is that it does not provide a detailed set of classes and properties to represent quality of service metrics. The QoS model needs to be extended to allow a precise characterization of each dimension.

6 CONCLUSIONS

We have shown the importance of quality of service management of workflow and introduced the concept of workflow quality of service (QoS). While QoS management has a high importance for organizations, current WfMSs and workflow applications do not provide full solutions to support QoS. Research is necessary in four areas: *specification, analysis algorithms and methods, monitoring tools, and mechanisms to control* the quality of service. In this paper, we focus on workflow QoS specification and the de-

velopment of algorithms and methods to calculate QoS. Based on the reviewed literature on quality of service in other areas, and accounting for the particularities of workflow systems and applications, we define a workflow QoS model, which includes four dimensions: time, cost, fidelity, and reliability. The use of QoS increases the added value of workflow systems to organizations, since non-functional aspects of workflows can be described. The specification of QoS involves fundamentally the use of an adequate model and the creation of realistic QoS estimates for workflow tasks. Once tasks have their QoS estimated, QoS metrics can be compute for workflows. Since this computation needs to be automatic we describe two methods for workflow QoS computation: analysis and simulation.

7 REFERENCES

- Ankolekar, A., et al, (2001), *DAML-S: Semantic Markup for Web Services*, in Proceedings of the International Semantic Web Working Symposium (SWWS), pp. 39-54.
- Bussler, C. (1998), *Workflow Instance Scheduling with Project Management Tools*, in 9th Workshop on Database and Expert Systems Applications DEXA'98, Vienna, Austria: IEEE Computer Society Press, pp. 753-758.
- Cardoso, J. (2002), *Stochastic Workflow Reduction Algorithm*. LSDIS Lab, Department of Computer Science, University of Georgia.
http://lsdis.cs.uga.edu/proj/meteor/QoS/SWR_Algorithm.htm. 2002.
- Cardoso, J. (2002), *Workflow Quality of Service and Semantic Workflow Composition*. Ph.D. Dissertation. Department of Computer Science, University of Georgia, Athens, GA.
- Dadam, P. Reichert, M. Kuhn, K. (2000), *Clinical workflows the killer application for process oriented information systems*, in 4th International Conference on Business Information System (BIS 2000), Poznan, Poland pp. 36-59.
- Damen, Z. et al, (2000), *Business-to-business E-Commerce in a Logistics Domain*, in The CAiSE'00 Workshop on Infrastructures for Dynamic Business-to-Business Service Outsourcing; Stockholm, Sweden.
- DAML-S, (2001), *Technical Overview - a white paper describing the key elements of DAML-S*.
- Eder, J. et al, (1999), *Time Management in Workflow Systems*, in BIS'99 3rd International Conference on Business Information Systems, Poznan, Polen: Springer-Verlag, pp. 265-280.
- Grefen, P. et al, (2000), *CrossFlow: Cross-Organizational Workflow Management in Dynamic Virtual Enterprises*. International Journal of Computer Systems Science & Engineering, 15(5): p. 227-290.
- Kao, B. Garcia Molina. H. (1993), *Deadline assignment in a distributed soft real-time system*, in Proceedings of the 13th International Conference on Distributed Computing Systems, pp. 428-437.
- Klingemann, J. Wäsch, J. Aberer, K. (1999), *Deriving Service Models in Cross-Organizational Workflows*, in Proceedings of RIDE - Information Technology for Virtual Enterprises (RIDE-VE '99), Sydney, Australia, pp. 100-107.
- Kochut, K.J., Sheth, A.P. Miller, J.A. (1999), *ORBWork: A CORBA-Based Fully Distributed, Scalable and Dynamic Workflow Enactment Service for METEOR*. Large Scale Distrib-

- uted Information Systems Lab, Department of Computer Science, University of Georgia: Athens, GA, USA.
- Marjanovic, O. Orłowska, M. (1999), *On modeling and verification of temporal constraints in production workflows*. Knowledge and Information Systems, 1(2).
- Marjanovic, O. Orłowska, M. (1999), *On modeling and verification of temporal constraints in production workflows*. Knowledge and Information Systems, 1(2), pp. 157-192.
- McCready, S., *There is more than one kind of workflow software*. Computerworld, 1992. November 2:86-90.
- Miller, J.A. Cardoso, J.S. Silver. G. (2002), *Using Simulation to Facilitate Effective Workflow Adaptation*, in Proceedings of the 35th Annual Simulation Symposium (ANSS'02). San Diego, California, pp. 177-181.
- Miller, J.A. Seila, A.F. Xiang, X. (2000), *The JSIM Web-Based Simulation Environment*. Future Generation Computer Systems: Special Issue on Web-Based Modeling and Simulation, 17(2): pp. 119-133.
- Miller, J.A., et al, (1998), *WebWork: METEOR2's Web-based Workflow Management System*. Journal of Intelligence Information Management Systems: Integrating Artificial Intelligence and Database Technologies (JIIS), 10(2): pp. 185-215.
- Pozewaunig, H. Eder, J. Liebhart, W. (1997), *ePERT: Extending PERT for workflow management systems*, in *First European Symposium in Advances in Databases and Information Systems (ADBIS)*. St. Petersburg, Russia, pp. 217-224.
- Sadiq, S. Marjanovic, O. Orłowska, M.E. (2000), *Managing Change and Time in Dynamic Workflow Processes*. The International Journal of Cooperative Information Systems, 9(1, 2), pp. 93-116.
- Son, J.H. Kim, J.H. Kim, M.H. (2001), *Deadline Allocation in a Time-Constrained Workflow*. International Journal of Cooperative Information Systems (IJCIS), 10(4), pp. 509-530.