Technical report, LSDIS Lab, Computer Science, Univiersity of Georgia, Athens GA USA, March 2002.

Workflow Quality of Service[1]

Jorge Cardoso, Amit Sheth and John Miller
LSDIS Lab, Department of Computer Science
University of Georgia
Athens, GA 30602 – USA
Amit@cs.uga.edu, http://lsdis.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. Management of such QoS directly impacts success of organizations participating in e-commerce. Organizations operating in modern markets require an excellent degree of quality of service management. Products and services must be available to customers with well-defined specifications. A good management of quality 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, Palaniswami et al. 1998; Kochut, Sheth et al. 1999) and applications (CAPA 1997; Anyanwu, Sheth et al. 1999;

---

[1] A paper based on this report will appear in Proceedings of the International Conf. on Enterprise Integration and Modeling Technology and International Enterprise Modeling Conference (ICEIMT/IEMC'02), Kluwer Publishers, April 2002.

Kang, Froscher et al. 1999; Hall, Miller et al. 2000; Luo 2000) made us aware that exiting workflow systems, both products and research prototypes, while providing a set of indispensable functionalities to manage and streamline business processes, they are missing an important requirement; the management of Quality of Service (QoS). Quality of service is an important issue for workflow systems. The international quality standard ISO 8402 (part of the ISO 9000 (ISO9000 2002)) describes quality as "*the totality of features and characteristics of a product or service that bear on its ability to satisfy stated or implied needs.*"

For organizations, being able to characterize workflows based on their QoS has three direct advantages. First, it allows organizations to translate more efficiently 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 monitoring of workflows based on QoS, and setting compensation strategies when undesired metrics are identified. It is essential that the services rendered follow customer specifications to ensure their expectation and satisfaction.

While QoS has been a major concern in the networking (Cruz 1995; Georgiadis, Guerin et al. 1996), real-time applications (Clark, Shenker et al. 1992) and middleware (Zinky, Bakken et al. 1997; Frlund and Koistinen 1998; Hiltunen, Schlichting et al. 2000) areas, few research groups have concentrated their effort to enhance workflow systems to support workflow Quality of Service management.

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 calculate QoS estimates. Finally, algorithms and methods need to be developed to compute workflow QoS. In workflows, quality metrics 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. For each dimension we present an approach to calculate QoS of workflow tasks. In section 3, we describe how QoS estimates for tasks can be created. Section 4 discusses two techniques to compute workflow QoS from task QoS. In section 5, we present a set of interesting QoS metrics which can be computed for a workflow. Section 6 and section 7 discuss related work in this area and give directions for future work. Finally, section 8 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*. Kobielus (1997) suggests that dimensions such as time, cost and quality should be the criteria that workflow systems should include and might benefit from. Qualitative characteristics specify the expected services offered by the system such as security and fault-tolerance mechanisms. QoS should be seen as an integral aspect of workflows, and therefore it should be integrated with workflow specifications.

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 (Kao and GarciaMolina 1993; Bussler 1998; Eder, Panagos et al. 1999; Marjanovic and Orlowska 1999; Dadam, Reichert et al. 2000; Sadiq, Marjanovic et al. 2000; Son, Kim 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, Panagos et al. 1999). The Crossflow project (Klingemann, Wäsch et al. 1999; Damen, Derks et al. 2000; Grefen, Aberer et al. 2000) is the one that most closely relates to our work. Not only time is considered, but also the cost associated with workflow executions. In Crossflow, the information about past workflow execution is collected in a log. From this information, a continuous-time Markov chain is derived.

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 (Garvin 1988; Stalk and Hout 1990; Rommel 1995) for organizations and quality of service for software systems (which include networking (Cruz 1995; Georgiadis, Guerin et al. 1996; Nahrstedt and Smith 1996), middleware areas (Zinky, Bakken et al. 1997; Frlund and Koistinen 1998; Hiltunen, Schlichting et al. 2000), and real-time applications (Clark, Shenker et al. 1992).) 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

According to Weikum (1999), information services QoS can be divided in three categories: system centric, process centric, and information centric. Based on previous studies and our experience in the workflow domain, we construct a QoS model that includes system and process categories. Out model is 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. 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. Therefore, *task response time* for a task *t* can be calculated as follows:

$$T(t) = DT(t) + PT(t)$$

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 *task realization cost.* The enactment cost (EC) is the cost associated with the management of the workflow system and workflow instances monitoring. The task realization cost (RC) is the cost associated with the runtime execution of the task.

$$C(t) = EC(t) + RC(t)$$

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 reflects how well a product is being produced and how well a service is being 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. Depending on its type, a task uses different strategies to set fidelity attributes.

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. Each workflow task structure has an initial state, an execution state, and two distinct terminating states. One of the states indicates that a task failed or was aborted while the other state indicates that a task is done or committed (Krishnakumar and Sheth 1995). This QoS dimension provides information concerning the relationship between the number of times the state done/committed is reached, and the number of times the failed/aborted state is reached. To describe task reliability we follow a discrete-time modeling approach. Discrete-time models are adequate for systems that respond to occasional demands such as database systems. We use the stable reliability model proposed by Nelson (1973), for which the reliability of a task *t* is $R(t) = 1$ - failure rate.

2.2 QoS Model and Web Services

Other researchers have also identified the need for a QoS process model. A good example is the DAML-S specification, which supplies an ontology to semantically describe business processes (as composition of Web services). The use of an ontology allows and facilitates process interoperability between trading partners involved in e-commerce activities. 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. The addition of concepts to represent the minimum, average, maximum, and the distribution function associated with dimension, such as cost and duration, will allow the implementation of algorithms for the automatic computation of QoS metrics of processes based on sub-processes' QoS metrics.

Let us try to better understand the impact and span of this problem. Workflows and processes are often composed of many of sub-processes (also known as composite process or network tasks). Processes can be represented using a hierarchical structure, where the root node corresponds to the main or top process, and the intermediate nodes and leaves correspond to sub-process and atomic processes (also known as atomic tasks) respectively. We believe that QoS metrics should be specified for the leaves (atomic processes) if at all possible. Then, using an appropriate algorithm, the QoS values of the leaves are used to compute QoS values for intermediate nodes (sub-processes) until the root node is reached. For organizations determining the QoS for an atomic process can be a complex procedure, but far more complex is to compute the QoS of a process composed by several sub-processes. Our work targets this computation, which based of atomic task QoS attributes, computes the quality of service for the process automatically. The use of such methodology to derive QoS for processes has one important requirement —the quality dimensions represented in the QoS model needs to be computable, *i.e.,* it must exist a function at each node of the hierarchical structure (networks) that can be applied to its children (atomic tasks). From this observation, we develop a QoS model for which all its dimensions are computable. We have investigated relevant work to determine which dimensions would be relevant to compose our wQoS model, and based on previous studies as well as our experience in the workflow domain, we have constructed a model composed of the following dimensions: time, cost, fidelity, and reliability. We hope this work will provide an input to the area of Web service specification related standards efforts, as well as E-services and process realization though composition of Web services.

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.

QoS for Tasks

Task QoS is initialize 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.

QoS for Transitions

The same way we estimate task QoS, can be used to estimate workflow transitions probabilities. The user initializes the transitions probabilities at design time. At runtime the probabilities are re-computed. When a workflow has never been executed, the values for the transitions are obviously taken from initial user specifications. When 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, Miller et al. 2002). This paper presents a stochastic workflow reduction algorithm (SWR) 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 (T), 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 queuing time and setup time. This is continued until only one atomic task (Kochut, Sheth 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. Invalid workflows contain design errors, such as non-termination, deadlocks, and split of instances (Aalst 1999). To compute QoS metrics, we use a set of six distinct reduction rules: (1) sequential, (2) parallel, (3) conditional, (4) fault-tolerant, (5) loop, and (6) network. As an illustration, we will show how reduction works for a sequence of tasks.

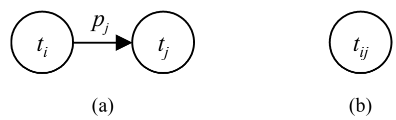

Figure 1 - Sequential system reduction

Reduction of a Sequential System. Two sequential service tasks t_i and t_j are reduced to a single task t_{ij}. In this reduction the incoming transitions of t_i and outgoing transition of tasks t_j are transferred to task t_{ij}.

In a sequential system $p_j = 1$. This reduction can only be applied if the following two rules are satisfied: a) t_i is not a *xor/and* split and b) t_j is not a *xor/and* join. These rules prevent this reduction from being applied in a parallel, conditional, and loop systems. To compute the QoS of the reduction the following formulae are applied:

Time :	$T(t_{ij}) = T(t_i) + T(t_j)$
Cost:	$C(t_{ij})= C(t_i) + C(t_j)$
Reliability:	$R(t_{ij}) = R(t_i) * R(t_j)$

4.2 Simulation Models

While analytical methods can be effectively used, another alternative is to utilize simulation analysis (Miller, Cardoso 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 METEOR WfMS and the JSIM simulation system (Nair, Miller et al. 1996; Miller, Nair et al. 1997; Miller, Seila et al. 2000). 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, 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 Workflow QoS Metrics of Interest

In this section, we list the workflow QoS metrics which are of interest to compute. The computation can be done at design time, before the execution of instances, or it can be done at runtime.

Workflow Time. Workflow time analysis measures the total time that instances spend in a workflow process. When a workflow is executed, instances enter the process, proceed through various tasks, and finally exit the process. The WfMS needs to constantly monitor and estimated the time remaining for instance termination. We show four important measurements for workflow time-based execution: *workflow response time*, *workflow delay time*, *minimum workflow response time*, and *workflow response time efficiency*.

- Workflow response time (T) is the total amount of time that a workflow instance spends in a workflow process before it finishes.
- Workflow delay time (DT) is the total amount of time that a workflow instance spends in a workflow while not being processed by a task.
- Minimum workflow response time (T_{min}) of a workflow is the time required for a workflow instance to be processed not having to account for any task delay time.
- Workflow response time efficiency (E) is the ratio of the minimum instance response time and the instance response time. It is instructive to compare these two measures, since instance efficiency measurement provides an indication of the time an instance is delayed during its execution and indicates to which degree a workflow process can be improved by reducing its response time.

Workflow Cost. Workflow cost (C) analysis measures the cost incurred with the execution of a workflow. A workflow has a cost which is equal to the sum of the cost of the tasks executed to complete a workflow. Cost-based workflows need to have their cost calculated so that managers can make sure that operations are within initial budgets.

Workflow Fidelity. Workflows fidelity (F) is computed based on the fidelity of the tasks in the workflow. The user defines a weighted function involving each task fidelity. A fidelity function is also defined to represent how well a task is carrying out its execution.

Workflow Reliability. Workflow reliability (R) corresponds to the likelihood that a workflow will perform for its users when the user demands it.

6 Related Work

The work found in the literature on quality of service for WfMS is limited. The Crossflow project (Klingemann, Wäsch et al. 1999; Damen, Derks et al. 2000; Grefen, Aberer et al. 2000) 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. Reichert (1998)and Dadam (2000) also recognize that time is an important aspect of workflow execution. With each workflow task, minimal and maximal durations may be specified. The system supports the specification and monitoring of deadlines. The monitoring system allows the notification of users when deadlines are going to be missed. It also checks if minimal and maximal time distances between tasks are followed according to initial specifications. Marjanovic and Orlowska (1999) describe a workflow model enriched with modeling constructs and algorithms for checking the consistency of workflow temporal constraints. Their work mainly focus on how to manage workflow changes while accounting for temporal constraints. Son (2001) present a solution for the deadline allocation problem based on queuing networks. Their work also uses graph reduction techniques, but applied to queuing theory. Although the work on quality of service for workflows is lacking, a significant amount of research on QoS has been done in the area of networking (Cruz 1995; Georgiadis, Guerin et al. 1996), real-time applications (Clark, Shenker et al. 1992) and middleware (Zinky, Bakken et al. 1997; Frlund and Koistinen 1998; Hiltunen, Schlichting et al. 2000).

Recently, in the area of Web services, researchers have also manifested an interest for QoS. The DAML-S (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. One current limitation of DAML-S' QoS model is that every composite process needs to have QoS metrics specified by the user.

7 Future Work

The workflow QoS model presented in this paper can be extended in two additional dimensions, which can be useful for particular types of workflow systems. The first one that can be included is the *security* dimension. Workflow systems and workflow applications face several security problems and dedicated mechanisms need to be present to increase the level of security (Fan 1999; Kang, Froscher et al. 1999; Miller, Fan et al. 1999). The second dimension, which can be added, is *maintainability*. Maintainability corresponds to the mean time to repair workflow failures; it is the average time spent to maintain workflow in a condition where they can perform their intended function.

8 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, prediction algorithms and methods*, *monitoring tools*, and *mechanisms to control* the quality of service. In this paper, we focus on workflow QoS specification and prediction. 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.

9 References

Aalst, W. M. P. v. d. (1999). Generic Workflow Models: How to Handle Dynamic Change and Capture Management Information. Proceedings of the Fourth IFCIS International Conference on Cooperative Information Systems (CoopIS'99), Edinburgh, Scotland.

Anyanwu, K., A. P. Sheth, et al. (1999). Healthcare Enterprise Process Development and Integration. Technical Report. Athens, GA, LSDIS lab, Department of Computer Science, University of Georgia.

Bussler, C. (1998). Workflow Instance Scheduling with Project Management Tools. 9 th Workshop on Database and Expert Systems Applications DEXA'98, Vienna, Austria, IEEE Computer Society Press.

CAPA (1997). Course Approval Process Automation (CAPA). Athens, GA, LSDIS Lab, Department of Computer Science, University of Georgia.

Cardoso, J., J. Miller, et al. (2002). Workflow Quality of Service: its Specification and Computation. Athens, GA, LSDIS Lab, Computer Science Department, Univ. of Georgia.

Clark, D., S. Shenker, et al. (1992). Supporting Real-Time Applications in an Integrated Services Packet Network: Architecture and Mechanism. Proceedings of ACM SIGCOMM.

Cruz, R. L. (1995). "Quality of service guarantees in virtual circuit switched networks." IEEE J. Select. Areas Commun. **13**(6): 1048-1056.

Dadam, P., M. Reichert, et al. (2000). Clinical workflows the killer application for process oriented information systems. 4th International Conference on Business Information System (BIS 2000), Poznan, Poland.

Damen, Z., W. Derks, et al. (2000). Business-to-business E-Commerce in a Logistics Domain. The CAiSE*00 Workshop on Infrastructures for Dynamic Business-to-Business Service Outsourcing, Stockholm.

DAML-S (2001). Technical Overview - a white paper describing the key elements of DAML-S.

Eder, J., E. Panagos, et al. (1999). Time Management in Workflow Systems. BIS'99 3rd International Conference on Business Information Systems, Poznan, Polen, Springer Verlag.

Fan, M. (1999). Security for the METEOR Workflow Management System. Department of Computer Science. Athens, GA, University of Georgia.

Frlund, S. and J. Koistinen (1998). "Quality-of-Service Specification in Distributed Object Systems." Distributed Systems Engineering Journal **5**(4).

Garvin, D. (1988). Managing Quality: The strategic and Competitive Edge. New York:, Free Press.

Georgiadis, L., R. Guerin, et al. (1996). Efficient network QoS provisioning based on per node traffic shaping. IEEE ACM Transactions on Networking.

Grefen, P., K. Aberer, et al. (2000). "CrossFlow: Cross-Organizational Workflow Management in Dynamic Virtual Enterprises." International Journal of Computer Systems Science & Engineering **15**(5): 227-290.

Hall, D., J. A. Miller, et al. (2000). Using Workflow to Build an Information Management System for a Geographically Distributed Genome Sequence Initiative. Athens, GA, University of Georgia, Department of Computer Science, LSDIS Lab.

Hiltunen, M. A., R. D. Schlichting, et al. (2000). Survivability through Customization and Adaptability: The Cactus Approach. DARPA Information Survivability Conference and Exposition (DISCEX 2000).

ISO9000 (2002). ISO9000 - http://www.iso.ch/iso/en/iso9000-14000/iso9000/iso9000index.html.

Kang, M. H., J. N. Froscher, et al. (1999). A Multilevel Secure Workflow Management System. Proceedings of the 11th Conference on Advanced Information Systems Engineering, Heidelberg, Germany.

Kao, B. and H. GarciaMolina (1993). Deadline assignment in a distributed soft realtime system. Proceedings of the 13th International Conference on Distributed Computing Systems.

Klingemann, J., J. Wäsch, et al. (1999). Deriving Service Models in Cross-Organizational Workflows. Proceedings of RIDE - Information Technology for Virtual Enterprises (RIDE-VE '99), Sydney, Australia.

Kobielus, J. G. (1997). Workflow Strategies, IDG Books Worldwide.

Kochut, K. J., A. P. Sheth, et al. (1999). ORBWork: A CORBA-Based Fully Distributed, Scalable and Dynamic Workflow Enactment Service for METEOR. Athens, GA, Large Scale Distributed Information Systems Lab, Department of Computer Science, University of Georgia.

Krishnakumar, N. and A. Sheth (1995). "Managing Heterogeneous Multi-system Tasks to Support Enterprise-wide Operations." Distributed and Parallel Databases Journal **3**(2): 155-186.

Luo, Z. (2000). Knowledge Sharing, Coordinated Exception Handling, and Intelligent Problem Solving to Support Cross-Organizational Business Processes. Department of Computer Science. Athens, GA, University of Georgia.

Marjanovic, O. and M. Orlowska (1999). "On modeling and verification of temporal constraints in production workflows." Knowledge and Information Systems **1**(2).

Miller, J. A., J. S. Cardoso, et al. (2002). Using Simulation to Facilitate Effective Workflow Adaptation. Proceedings of the 35th Annual Simulation Symposium (ANSS'02), San Diego, California.

Miller, J. A., M. Fan, et al. (1999). Security for the METEOR Workflow Management System. Athens, GA, Department of Computer Science, University of Georgia**:** 33.

Miller, J. A., R. Nair, et al. (1997). JSIM: A Java-Based Simulation and Animation Environment. Proceedings of the 30th Annual Simulation Symposium, Atlanta, GA.

Miller, J. A., D. Palaniswami, 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): 185-215.

Miller, J. A., A. F. Seila, et al. (2000). "The JSIM Web-Based Simulation Environment." Future Generation Computer Systems: Special Issue on Web-Based Modeling and Simulation **17**(2): 119-133.

Nahrstedt, K. and J. M. Smith (1996). "Design, Implementation and Experiences of the OMEGA End-point Architecture." IEEE JSAC **14**(7): 1263-1279.

Nair, R., J. A. Miller, et al. (1996). A Java-Based Query Driven Simulation Environment. Proceedings of the 1996 Winter Simulation Conference, Colorado, CA.

Nelson, E. C. (1973). A Statistical Basis for Software Reliability, TRW Software Series.

Pozewaunig, H., J. Eder, et al. (1997). ePERT: Extending PERT for workflow management systems. First European Symposium in Advances in Databases and Information Systems (ADBIS), St. Petersburg, Russia.

Reichert, M. and P. Dadam (1998). "ADEPTflex -- Supporting Dynamic Changes of Workflows Without Losing Control." Journal of Intelligent Information Systems - Special Issue on Workflow Managament **10**(2): 93-129.

Rommel, G. (1995). Simplicity wins: how Germany's mid-sized industrial companies succeed. Boston, Mass, Harvard Business School Press.

Sadiq, S., O. Marjanovic, et al. (2000). "Managing Change and Time in Dynamic Workflow Processes." The International Journal of Cooperative Information Systems **9**(1, 2).

Son, J. H., J. H. Kim, et al. (2001). "Deadline Allocation in a Time-Constrained Workflow." International Journal of Cooperative Information Systems (IJCIS) **10**(4).

Stalk, G. and T. M. Hout (1990). Competing against time: how timebased competition is reshaping global markets. New York, Free Press.

Weikum, G. (1999). Towards guaranteed quality and dependability of information systems. Proceedings of the Conference Datenbanksysteme in Buro, Technik und Wissenschaft, Freiburg, Germany.

Zinky, J., D. Bakken, et al. (1997). "Architectural Support for Quality of Service for CORBA Objects." Theory and Practice of Object Systems **3**(1): 1-20.