Using Simulation to Facilitate Effective Workflow Adaptation

John A. Miller
Jorge Cardoso
Gregory Silver

Computer Science Department
University of Georgia
Athens, GA 30602, U.S.A.

Abstract

In order to support realistic real-world processes, workflow systems need to be able to adapt to changes. Detecting the need to change and deciding what changes to carry out are very difficult. Simulation analysis can play an important role in this. It can be used in tuning quality of service metrics and in exploring "what-if" questions. Before a change is actually made, its possible effects can be explored with simulation. To facilitate rapid feedback, the workflow system (METEOR) and simulation system (JSIM) 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. Fortunately, modern Web technology (e.g., XML, DTD, XSLT) make this relatively straightforward. The utility of using simulation in adapting a workflow is illustrated with an example from a genome workflow.

1 Introduction

Workflow systems are used to automate/support real-world processes in organizations [1, 2, 3]. Examples of real-world processes for which workflow automation has been found to be helpful include outpatient healthcare treatment, equipment procurement, course proposal processing, and data-intensive scientific experimentation. A workflow can be represented as a directed graph (digraph) consisting of a set tasks (nodes) that are interconnected with control/data flow edges (other representations such as Petri nets are also used). The Large-Scale Distributed Information Systems (LSDIS) Lab at the University of Georgia has been carrying out research on Workflow Management Systems (WfMSs) since the middle 1990's [4, 5, 6, 7] under the METEOR project.

Workflow Management Systems have met with some success. One reason for the limited success is that real-world processes are very complex. Furthermore, since real-world processes in business, engineering and science are ever-changing, a WfMS should support such change. The LSDIS Lab has a major project underway to provide a next-generation WfMS that will hopefully be able to keep up with the changeable nature of real-world processes. The latest METEOR research prototype, called OrbWork [7], supports dynamic changes to workflows. OrbWork supports dynamic changes which allow a human designer to rapidly change a running workflow without shutting it down and starting over. That is, while there are active workflow instances running changes can be made to the workflow: task substitution, task insertion, task deletion, flow edge change, flow edge insertion, or flow edge deletion, as well as combinations of these.

Our next step is to make OrbWork more adaptive, so that changes are easier to make. Adaptation is necessary because real-world processes change and if the workflow does not keep up these changes, its value will diminish. If the WfMS does not support adaptation, keeping the workflow current can become very difficult. Therefore, our most recent research direction is to add adaptive capabilities to OrbWork. OrbWork can change which is good, but currently (i) detecting the need for change, and (ii) the type of change to be made is purely up to the humans managing the workflow. These capabilities need to be at least partially automated, so that adaptation can be done in a computer-aided fashion and eventually, according to some, fully automated.

Because of the complexity of adapting workflows, simulation can be used to check the desirability of the change. In addition, desirability needs to be quantified. This is done by introducing Quality of Service (QoS) metrics into workflows.

In our project, these capabilities involve a loosely-coupled integration between the METEOR WfMS and the JSIM simulation system [8, 9, 10]. Since in both systems, models are represented as directed graphs and XML documents are used for storing designs, interoperation is facilitated. Furthermore, both conceptualize entities flowing through a network (digraph) of nodes. Workflow is concerned with scheduling and transformations that take place in tasks, while simulation is mainly concerned with system performance.

METEOR and JSIM provide synergy in two directions: (1) Given a workflow design (either an original design or a proposed adaptation) a simulation model can be created for it. Studying the performance profiles of the workflow design using simulation will provide feedback to the design process for improving the workflow design before developing or deploying the workflow or carrying out the adaptation. (2) Once a workflow is deployed and running, performance data can be collected to refine or validate the simulation model. A validated simulation model will be very useful for future adaptations of the workflow [11].

The rest of this paper is organized as follows: In section 2, we will discuss quality of service metrics and how they can be used to guide adaptation. The use of simulation analysis to estimate the desirability of particular adaptations is presented in section 3. Section 4, highlights some of the issues involved in integrating the METEOR and JSIM systems. An illustrative example is given in section 5 in which one of the tasks needs to be changed. The reader is walked through this process, step by step. Finally, conclusions and future work are given in section 6.

2 Quality of Service

In order for adaptation to proceed in a positive direction, cost models should be used. If models are not formally part of the software systems, then they must be used at least implicitly by humans modifying the workflow. Our project uses cost models explicitly in either a computer-aided or fully automated mode. Each task may have a cost model included in it. Let us consider the execution of task τ that begins at time t_1 and completes at time t_2. This execution will take input and produce output as well as possibly modifying the state of the task. Overall quality will be decomposed into three dimensions (quality metrics).

1. $time = t2 - t1$
2. $cost = f_c(\tau, output)$
3. $quality = f_q(\tau, output)$

The cost and quality functions are attached to the task by the task designer. If the task was designed by modifying an existing task found in a task repository, then the old model may be reused or modified.

3 Simulation Analysis

Since workflow adaptations may change quality in either direction, safe guards need to be used. The new workflow should be (i) analyzed structurally [12], (ii) only a combination of well-defined dynamic changes should be permitted [7], and (iii) simulated to estimate how the three quality metrics will change.

3.1 Modeling of Tasks

To simulate task execution, service times are drawn from a random variate. This distribution is specified by the simulation model designer and can be adjusted based upon data collected from running workflows. Statistics are collected on task execution/service times as well as waiting times. Task instances may be queued waiting on the availability of limited resources. The total time for a workflow instance is the sum of the waiting time and service time.

3.2 Modeling of Workflow

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). The directed graph (digraph) consists of a nonempty set of nodes and edges with the following properties:

1. There are two types of nodes (AND nodes and XOR nodes). Each node also has an associated task type (*e.g.,* transactional, nontransaction, human). A Boolean property indicates whether workflow instances can be queued at the task.
2. One of the nodes in the digraph must be designated to be a START node and one must be designated to be a STOP node. The START node must have no incoming edges, while the STOP node must have no outgoing edges.
3. The digraph must be weakly connected.
4. Edge labels indicate the probability of edges being selected as the outgoing edge. These probabilities should be chosen by analyzing the conditions on implementation oriented workflow specifications or data collected from running actual workflow instances. For an XOR node, one outgoing edge should be selected, so these probabilities must add up to one. Edges emanating from AND nodes should not be labeled, since all outgoing edges will selected.

3.3 Simulation of Workflows using JSIM

In order to simulate METEOR workflows, we are enhancing the JSIM Web-Based Simulation System [8, 9, 10] In JSIM, simulation entities flow through a digraph consisting of the following types of nodes.

`Source`	produces entities with random times
`Server`	provides service to entities
`Facility`	inherits from Server, adds a waiting queue
`Signal`	alters number of service units in a server(s)
`Sink`	consumes entities and records statistics

Table 1: Nodes in JSIM

These nodes are connected together with `Transports` which move entities from one node to the next. These edges provide smooth motion of entities when a simulation model is animated. These edges are labeled with branching probabilities.

The principal enhancement is to add a property to nodes to indicate whether they are AND or XOR nodes. Currently, all nodes are XOR nodes (an entity) will only follow one outgoing edge. We will implement AND nodes by cloning entities, so that they will follow all outgoing edges. Conversely, all entities coming into an AND node will be merged into one. A customizable join rule will indicate how the merger is to be carried out. Another complexity introduced by adding AND nodes is the issue of graph well-formedness [12] (*e.g.*, AND splits (edges coming out of AND nodes) should be matched by AND joins (edges coming into AND nodes). AND nodes are differentiated from XOR nodes by having more rounded versions of the node icons.

With the addition of AND nodes, the mapping of a workflow digraph to a simulation digraph is straightforward. A METEOR START, STOP task will be mapped to a JSIM `Source`, `Sink` node, respectively. A METEOR human task will be mapped to a a JSIM `Facility` with the number of service units equal to the number of human participants carrying out the task and feeding off the same worklist. A METEOR transactional/nontransactional task will be mapped to a JSIM `Facility` with the number of service units equal to the number of processors available to execute the task. These default mappings can be customized (*e.g.*, a nontransactional task that does not allow requests to be queued should be mapped to a JSIM `Server`). Each edge in the METEOR digraph will be mapping to a corresponding edge in the JSIM digraph. In METEOR, edges are labeled with the data type of objects flowing along the edge. In the case of XOR nodes, they are also labeled with Boolean expressions (the first one that evaluates to true will be the edge selected). In the current of version of JSIM, data flow must be handled by custom coding. A Boolean expression is mapped to the probability that the condition will evaluate to true and none of the preceding conditions will evaluate to true.

Utilizing a Process-Interaction (PI) simulation engine, JSIM implements each simulation entity as a thread whose run method advances the entity from the `Source` through the digraph and finally to the `Sink` where it is destroyed.

3.4 Change Evaluation

After a task finishes executing, its time, cost and quality metrics will be computed. These values will be sent to the monitor (and saved in the monitor's log). Each of these three metrics will be compared to *dual threshold* values. If the value is below the bottom threshold, an appropriate exception will be thrown. If it is above this threshold, but below the top threshold, a warning will be sent to the monitor. If the value is above the top threshold, the execution is considered to be satisfactory.

If an exception is thrown, it will be handled by METEOR's exception handling facility. Several options exist here such as retry the task, use an alternative task or adapt the workflow. As warnings are accumulated by the monitor, a pattern may emerge that indicates an adaptive change could be helpful. Such a pattern could be detected by a human observing the workflow monitor or by a data mining agent examining the monitor's log. The detecting process consists of identifying and classifying a pattern of substandard quality. After this, possible corrective actions are determined by taking the top n template matches. Details of pattern detection and template matching are given in [12].

Simulation is useful in setting the thresholds and in creating templates that suggest corrective actions (*i.e.*, workflow adaptations). Simulation allows "what-if" questions to be considered safely (*i.e.*, what happens if one makes this change to the workflow).

4 System Integration

The integration of METEOR with JSIM is depicted in Figure 1. In order to facilitate rapid simulation of workflow adaptation, it is necessary to translate workflow design specifications into simulation specifications. The METEOR designer [12] saves workflow designs as XML documents stored in an XML Model Repository. Such a design can be retrived from the repository and translated into JSIM simulation specifications. The translation is simplified since both systems represent their models as digraphs with AND and XOR nodes. A default mapping from a workflow node to a simulation

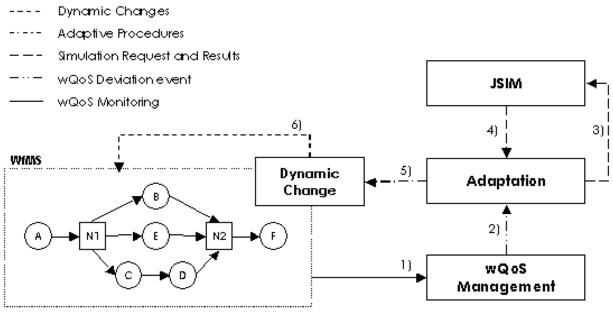

Figure 1: Overall System Architecture

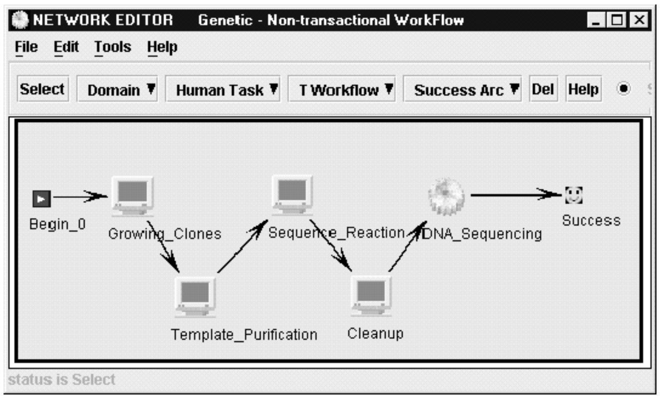

Figure 2: Genome Sequencing Workflow

node is carried out based on the node type. This default can be adjusted using the JSIM model design tool before the simulation is actually run.

Workflow designs follow their own Document Type Definition (DTD), while the simulation designs follow a different DTD. Consequently, a workflow design document must be mapped to a simulation design document. This is accomplished using an XSLT specification.

5 Case Study: A Genomics Workflow

In this section, we illustrate the steps that take place in workflow adaptation. The reference workflow is a Genome Sequencing Workflow which is currently under development by the LSDIS Lab in conjunction with, Dr. Jonathan Arnold, a professor from the Genetics Department.

The Genomic Sequencing Workflow is composed of several tasks at the top level of the hierarchical workflow design (see [12] for details). The main tasks include `Prepare Sample`, `Prepare Clones`, `Sequencing`, and `Sequence Processing`. For our case study, we only are interested in the `Sequencing` network task (subworkflow). This task has a first phase to grow clones and a second phase to use DNA sequencing machines to read each biochemical letter (A, G, C or T) of a cloned DNA fragments. The workflow is represented in Figure 2.

Let us turn our attention to the `Cleanup` task. A specialized laboratory technician carries out this task, which cleans DNA material using a precipitation method. The quality metrics for the `Cleanup` task are:

1. task cycle time = waiting time + labor time = 1.0 h + 1.5 h = 2.5 h

2. task cost = consumables + labor rate * length of labor = \$0.6 + \$20 * 1.5 h = \$30.6

3. task quality = 95%

Now suppose that during instance execution the quality metrics repeatedly drop below the top thresholds, because the lab technician has an infection and *E. coli* bacteria is involuntarily added to the sample to cleanup. This situation causes the purity of the DNA to drop significantly and the task cleanup shows a quality factor inferior to 95%. As a result, the WfMS detects this insufficient quality for the sample being sequenced and sends warning messages to the monitor.

The flurry of warning messages should become apparent to someone monitoring the workflow. (For a discussion of how an agent can discover the same thing see [12].) By template matching, possible adaptive changes are suggested along with a figure of merit indicating the closeness of the match. In this case we assume that the first choice is task substitution. The WfMS searches the available tasks in the repository to find a substitute task with predicted values for the quality metric that are more desirable. Let us suppose the system finds a task in the repository, which also cleans up DNA samples, but using a different chemical process. Instead of cleaning DNA by precipitation, a ready-to-use kit is utilized. The task has been used in the past, therefore statistical values on past behavior has been recorded. The recorded values are shown below.

1. task cycle time = waiting time + labor time = 0 h + 0.25 h = 0.25 h

2. task cost = consumables + labor rate * length of labor = \$40 + \$20 * 0.25 h = \$45

3. task quality = 98%

According to these statistics, the potential substitute task will improve the quality of the cleanup process (from 95% to 98%), and additionally reduce the time necessary to complete the task (from 150 to 15 min). Unfortunately, the use of this task increases the cost of

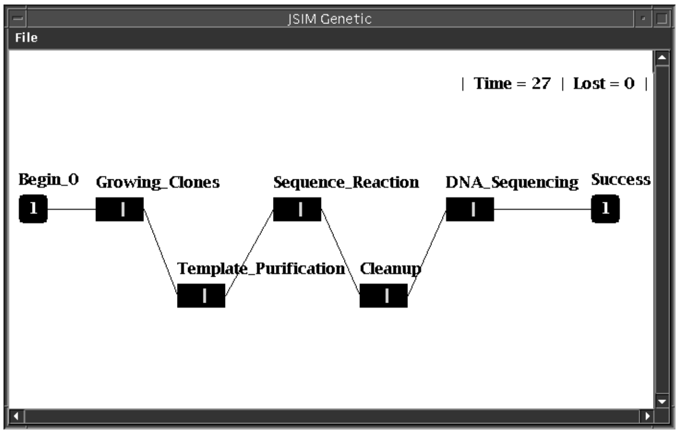

Figure 3: Animation Snapshot

the overall workflow by nearly $15. Since the determination of the overall gain of adapting the workflow using this new task is ambiguous and the statistical estimates were not collected for this particular use, a simulation of the proposed workflow adaptation is performed before committing to any change.

In order to carry out this 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 as shown in Figure 3. Statistical results are collected and displayed which indicate how effective the adaptation is.

6 Conclusions

This paper has shown how simulation can be useful for supporting adaptive workflows. One can ask "what-if" questions via a simulation as opposed to actually trying the change and hoping it has no bad consequences.

Because of analogs in the conceptual frameworks of both METEOR and JSIM, interoperability is facilitated. Since design tools for both systems use XML for saving designs in a repository, design specifications can be translated using XSLT. Currently, translation is in the direction from workflow specification to simulation specification. This is because that is what is needed and the fact that the simulation specification contains fewer details (*i.e.*, it is more abstract).

The example genomics workflow illustrates how important quality metrics are to (i) deciding whether a workflow should be adapted and (ii) what change should actually be made. These metrics are important for both the workflow and the simulation.

References

(1) A.P. Sheth and M. Rusinkiewicz, "On Transactional Workflows," *Data Engineering Bulletin*, 16(2), Jun 1993.

(2) D. Georgakopoulos, M. Hornick and A.P. Sheth, "An Overview of Workflow Management: From Process Modeling to Workflow Automation Infrastructure," *Distributed and Parallel Databases*, 3(2):119–154, April 1995.

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

(4) A.P. Sheth, K.J. Kochut, J.A. Miller, D. Worah, S. Das, C. Lin, D. Palaniswami, J. Lynch and I. Shevchenko, "Supporting State-Wide Immunization Tracking using Multi-Paradigm Workflow Technology," *Proc. of the 22nd. Intnl. Conference on Very Large Data Bases*, pp. 263-273, Bombay, India, Sep 1996.

(5) J.A. Miller, A.P. Sheth, K.J. Kochut and X. Wang, "CORBA-based Run-Time Architectures for Workflow Management Systems," *Journal of Database Management*, 7(1):16–27, Winter 1996.

(6) J.A. Miller, D. Palaniswami, A.P. Sheth, K.J. Kochut and H. Singh, "WebWork: METEOR_2's Web-Based Workflow Management System," *Journal of Intelligent Information Systems*, 10(2):185–215, Mar 1998.

(7) K.J. Kochut, A.P. Sheth and J.A. Miller, "Optimizing Workflow," *Component Strategies*, 1(9):45–57, Mar 1999.

(8) R. Nair, J.A. Miller and Z. Zhang, "A Java-Based Query Driven Simulation Environment," *Proc. of the 1996 Winter Simulation Conference*, pp. 786–793, Coronado, CA, Dec 1996.

(9) J.A. Miller, R. Nair, Z. Zhang and H. Zhao. "JSIM: A Java-Based Simulation and Animation Environment," *Proc. of the 30th Annual Simulation Symposium* pp. 786–793, Atlanta, GA, Apr 1997.

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

(11) J.A. Miller, A.P. Sheth, K.J. Kochut, X. Wang and A. Murugan, "Simulation Modeling Within Workflow Technology," *Proc. of the 1995 Winter Simulation Conference*, pp. 612–619, Arlington, VA, Dec 1995.

(12) J. Cardoso, J.A. Miller and A.P. Sheth, "Workflow Quality of Service: Its Specification and Computation," The VLDB Journal, 2002. (in preparation)