© 2005 IEEE. Personal use of this material is permitted. However, permission to reprint/republish this material for advertising or promotional purposes or for creating new collective works for resale or redistribution to servers or lists, or to reuse any copyrighted component of this work in other works must be obtained from the IEEE.

Probabilistic Calculation of Execution Intervals for Workflows

Johann Eder and Horst Pichler
Institute for Informatics-Systems, University of Klagenfurt, Austria
[eder|pichler]@uni-klu.ac.at

Abstract

The comprehensive treatment of time and time constraints is crucial in designing and managing business processes. Process managers need tools that help them predict execution durations, anticipate time problems, pro-actively avoid time constraint violations, and make decisions about the relative process priorities and timing constraints when significant or unexpected delays occur. Variations of activity durations and branching decisions at run-time make it necessary that we treat time management in workflows in a probabilistic way. Therefore we introduce the notion of probabilistic time management and discuss the application of this new concepts for workflow design as well as time aware, predictive and proactive workflow execution management.

1 Introduction

Systems for business process automation, like workflow management or enterprise resource planning systems, are used to improve processes by automating tasks and getting the right information to the right place for a specific job function. As automated business processes often span several enterprises, the most critical need in companies striving to become more competitive is a high quality of service, where the expected process execution time ranks among the most important quality measures [7]. Additionally it is a necessity to control the flow of information and work in a timely manner by using time-related restrictions, such as bounded execution durations and absolute deadlines, which are often associated with process activities and sub-processes [3]. However, arbitrary time restrictions and unexpected delays can lead to time violations, which typically increase the execution time and cost of business processes because they require some type of exception handling [9].

Although currently available commercial products offer sophisticated modelling tools for specifying and analyzing workflow processes, their time management functionality is still rudimentary and mostly restricted to monitoring of constraint violations and simulation for process reengineering purposes [1, 6]. In research several attempts have been made to provide solutions to advanced time management problems (e.g. [1, 2, 6, 8]). Most of them suffer from the vagueness of information which stems mainly from two aspects: The duration of a task can vary greatly without any possibility of the workflow system to know beforehand. The second is that in a workflow different paths may be chosen with decisions taking place during the execution.

The main motivations for our probabilistic approach are: a) the improvement of estimations about the (remaining) duration of a workflow (predictive time management), and b) to make forecasts for the likelihood of deadline misses and automatically trigger escalation-avoiding actions if a possible future deadline violation has been detected (proactive time management). Our calculation algorithms utilize the knowledge about the control flow of a workflow and stochastic information about the uncertainties mentioned above.

2 Timed Workflow Graph

A workflow can be defined as directed acyclic graph, which is a collection of *nodes* and *edges* between nodes. Edges determine the execution sequence of nodes, thus a successor can start if its predecessor(s)

Proceedings of the 12th International Symposium on Temporal Representation and Reasoning (TIME'05)
1530-1311/05 $20.00 © 2005 **IEEE**

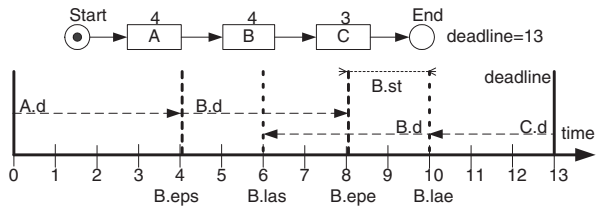

Figure 1. Implicit Time Properties

are finished. A node can be of type *activity*, which corresponds to individual tasks of a business process, or a control node (e.g. start, and-split, and-join, etc.). Additionally time properties can be attached to each node. Time properties are either explicit, if defined by the workflow designer, or implicit, if they follow implicitly from the workflows structure and explicit time properties [6]. We use a linear time model, where time is discrete with a universal predefined chronon, called basic time unit. The time line starts at 0, which denotes the start time of the workflow. All other points in time are declared or calculated relative to this start time.

Figure 1 visualizes a workflow consisting of three activities executed in sequence. Explicit time properties are the estimated duration of activities in basic time units, which are $A.d = 4$, $B.d = 4$ and $C.d = 3$ and a deadline of $\delta = 13$, stating that the overall workflow execution must not exceed 13 time units. Based on this information four implicit time properties can be calculated for each node (e.g. for activity B): a) Considering the sum of durations of preceding activities the *earliest possible start* of activity B is $B.eps = 4$. b) The according *earliest possible end* is $B.epe = B.eps + B.d = 8$. c) To take the deadline of 13, into account, the point of view has to be reversed, now starting from the end of the workflow. By subtracting the durations of succeeding activities from the deadline, the *latest allowed end* $B.lae$ of activity B is determined as $B.lae = 13 - 3 = 10$. That means if B ends at 10 it is still possible to reach the overall deadline of 13. d) Analogously the *latest allowed start time* is $B.las = B.lae - B.d = 6$.

This information can be utilized to specify a valid time interval for the execution of each activity, which ensures no time violations. E.g. we can state that B can not start before 4 and must end until 10 in order to hold the deadline. Additionally we can state that the expected duration of the workflow is equal to the sum of all activity durations, which is equal to $C.epe = 11$.

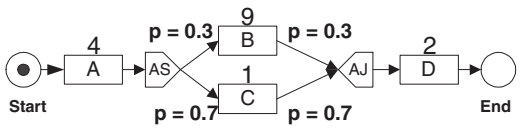

Figure 2. Workflow with or-structure

3 Probabilistic Timed Workflow Graph

The example presented above is rather simplistic, as some essential problems have not been addressed: a) The expected execution duration of a node is represented as a single average value (without variance, which is unusable for administrative workflows with human participants), b) in and-structures (parallel execution of nodes) the longest of all concurrently executed routes must be considered, and c) in or-structures (conditional execution of several alternative nodes) different paths may be chosen with decisions taking place during the execution, whose outcome we can not know when modelling the workflow. Therefore, it is impossible to unambiguously determine implicit time constraints or the overall workflow duration, especially if the workflow-graph contains complex control structures. To tackle these problems we introduced the *probabilistic timed graph*, which is an extension of the above presented basic model augmented with branching probabilities and time histograms.

Consider the graph in Fig. 2: The or-split after A forces a decision during the execution of the workflow. One of two possible routes will be chosen, therefore it is impossible to calculate one scalar value for the earliest possible start time of D. According to the branching probabilities (expert estimations or extracted from the log) D will start at 13 with a probability of 30% or at 5 with a probability of 70%. The same can analogously be stated for the latest allowed end time of A (as latest allowed times are calculated in a reversed fashion starting from the deadline defined on the last activity). Additionally a workflow modeler will rather use distribution functions to represent activity durations than single scalar values.

Therefore we introduced the concept of a time histogram, which is defined as a set of tuples (p, t) where p is the probability and t is the according time value. Time histograms are used to represent time properties in the form of probability distributions. A graph where

Proceedings of the 12th International Symposium on Temporal Representation and Reasoning (TIME'05)
1530-1311/05 $20.00 © 2005 **IEEE**

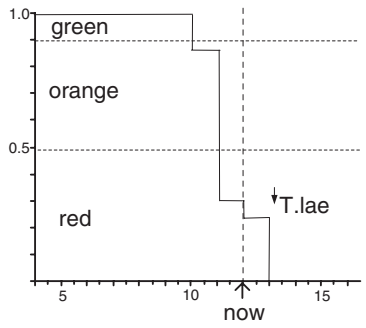

Figure 3. Traffic light model for activity T

each time property is represented by a time histogram is called *Probabilistic Timed Workflow Graph*. Details about time histograms, how to cumulate, interpret and query them and how to calculate the probabilistic timed model, considering different types of control nodes, is explained in [5]. How to cope with large histograms by compressing them and how to deal with blocked loop structures is explained in [4].

4 Application areas

We differ between predictive and proactive time management applications. *Predictive time management* is used to provide users (customers) with predictions about a expected execution durations or the likelihood of coming activity assignments (scheduling forecasts).

Proactive time management tries to asses the current situation, corresponding to possible future time violations, e.g. deadline misses. Time histograms are the basis for simple but effective escalation warning mechanisms, using an adaption of the *traffic light model* introduced in [3]: Two thresholds must be defined on the histogram representing the latest allowed end time of an activity. The first determines the workflows state change from green (ok) to yellow (warn) and the second one determines the state change from yellow to red (alarm). As long as the workflows state is green everything is ok, if the state changes something has to happen. Example: Assume that activity T just finished, at a point in time denoted by $now = 12$. Figure 3 shows the descending cumulated time histogram for $T.lae$ with two thresholds defined at 90% and 50%. Applying a selection-operation on the histogram yields a probability of 30% that the workflow can be finished without violating the overall deadline. This switches the status to red (for further details we refer to [5]).

According to the new state different escalation actions can be invoked, e.g. for orange this could be skipping unnecessary (optional) tasks. At status red an early escalation of this workflow might be invoked which aims at avoiding useless resource consumption of future activities (see also [9]). The important contribution of this approach is that threshold values can be expressed as the probability of a deadline violation.

5 Conclusion

Probabilistic time management will produce major advantages like better scheduling decisions and improved escalation strategies for workflow execution, as well as it provides the means to implement quality insurance components based on probabilistic time properties. The integration of probabilistic time management applications into workflow environments are subject of ongoing research.

References

[1] C. Combi and G. Pozzi. Temporal conceptual modelling of workflows. LNCS 2813. Springer, 2003.

[2] P. Dadam and M. Reichert. The adept wfms project at the university of ulm. In Proc. of the 1st European Workshop on Workflow and Process Management (WPM'98). Swiss Federal Institute of Technology (ETH), 1998.

[3] J. Eder and E. Panagos. Managing Time in Workflow Systems. Workflow Handbook 2001. Future Strategies Inc. Publ. in association with Workflow Management Coalition (WfMC), 2001.

[4] J. Eder and H. Pichler. Duration Histograms for Workflow Systems. In Proc. of the Conf. on Engineering Information Systems in the Internet Context 2002, Kluwer Academic Publishers, 2002.

[5] J. Eder and H. Pichler. Probabilistic Workflow Management. Technical report, Universität Klagenfurt, Institut für Informatik Systeme, 2005.

[6] J. Eder, E. Panagos, and M. Rabinovich. Time constraints in workflow systems. LNCS 1626. Springer, 1999.

[7] M. Gillmann, G. Weikum, and W. Wonner. Workflow management with service quality guarantees. In Proc. of the 2002 ACM SIGMOD Int. Conf. on Management of Data. ACM Press, 2002.

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

[9] E. Panagos and M. Rabinovich. Predictive workflow management. In Proc. of the 3rd Int. Workshop on Next Generation Information Technologies and Systems, Neve Ilan, ISRAEL, 1997.

Proceedings of the 12th International Symposium on Temporal Representation and Reasoning (TIME'05)
1530-1311/05 $20.00 © 2005 **IEEE**