DURATION HISTOGRAMS FOR WORKFLOW SYSTEMS

Johann Eder and Horst Pichler
Department of Informatics-Systems
University of Klagenfurt A-9020 Klagenfurt, Austria
eder@isys.uni-klu.ac.at, hpichler@edu.uni-klu.ac.at

Abstract Web-based interorganizational workflows need efficient time management. Variations of activity durations and branching distributions on or-split nodes make it necessary that we treat time management in workflows in a probabilistic way. We introduce the notion of duration histograms for capturing the available temporal information about workflow execution, define the necessary operations for computing timed execution plans for workflows and discuss the application of this new concepts for workflow design as well as time aware workflow execution management.

Keywords: Workflow management system, time plans, temporal constraints

1. Introduction

One of the prominent albeit obviously difficult application areas of web services is the support of business processes both within an organization as well as business processes spanning several organizations like supply chains. Today, the most critical need in companies striving to become more competitive is the ability to control the flow of information and work throughout the enterprise in a timely manner. Consequently, time-related restrictions, such as bounded execution durations and absolute deadlines, are often associated with process activities and sub-processes. However, arbitrary time restrictions and unexpected delays could lead to time violations. Typically, time violations increase the cost of business processes because they require some type of exception handling [12]. Therefore, the comprehensive treatment of time and time constraints is crucial in designing and managing business processes. Process managers need tools that help them anticipate time problems, pro-actively avoid time constraints violations, and make decisions about the relative process priorities and timing constraints when significant or unexpected delays occur.

Workflow management systems (WFMSs) improve business processes by automating tasks, getting the right information to the right place for a specific job function, and integrating information in the enterprise [4, 5, 8, 14]. Many workflow systems use the web as transportation media for process data. Although currently available commercial workflow products offer sophisticated modeling tools for specifying and analyzing workflow processes, their time management functionality is rudimentary.

In research some attempts have been made to provide solutions of this problem (e.g. [13, 6, 7, 10, 11, 12, 1, 3, 2, 9]). Most of them suffer from the uncertainty of time information in processes. This vagueness 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. Some approaches try to address this problem by introducing time intervals (best and worst case), or suppose some kind of distribution of duration values.

The contribution of this paper is the introduction of a new structure called *duration histograms* for representing time information and to make use of the different probabilities for different branches at split-nodes to improve the estimates for the duration of workflows and for the likelihood of deadline misses.

2. Workflow model

We define the rather generic workflow model we use in the rest of this paper and introduce *Duration Histogram* as a structure for representing probabilistic information about the duration of activities and processes.

Essentially, a workflow is a collection of *activities,* and *dependencies* between activities. Activities correspond to individual steps in a business process. Dependencies determine the execution sequence of activities and the data flow between them. Activities can be executed sequentially, repeatedly in a loop, or in parallel (and splits) or conditional (or-splits). Consequently, a workflow can be represented by an directed acyclic graph, where nodes correspond to activities and edges correspond to dependencies between activities. (Loops are introduced as repetitions of workflow graphs.)

Figure 1 shows an example workflow schema. Each activity is represented by a rectangle that holds its unique name and its estimated duration (given in any desired time-unit TU). A is the start-activity and T is the final activity. B will be executed when activity A is finished.

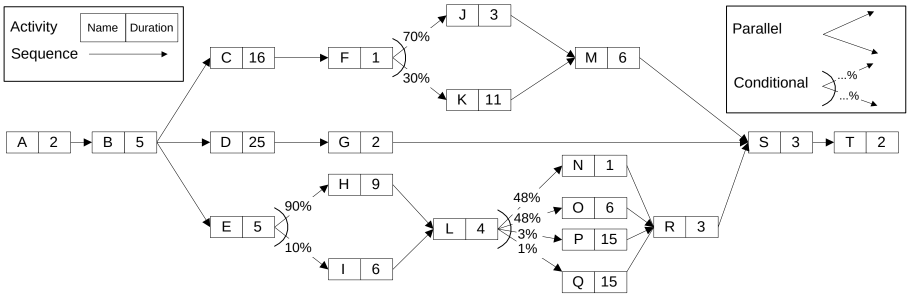


Figure 1. Example workflow process schema

The 3 routes after B (and-split) will be processed concurrently. That means, that the workflow continues with S (and-join) not until M, G and R are finished. Conditional branches (*or-split*) exist after E, F and L. In this cases only one specific path will be chosen, which for example results in 8 different execution-routes between E and T.

Additionally, the model contains statistically weighted values for each conditional branch. These probabilities can be defined by administrator estimations or average values from past executions. Regarding the example, H will be executed after E in 9 out of 10 cases and O will be executed after L in 4.8 out of 10 cases. Thus the probability that a workflow-instance will execute the path from E via H and O to T is $0.9 \cdot 0.48 = 0.432$.

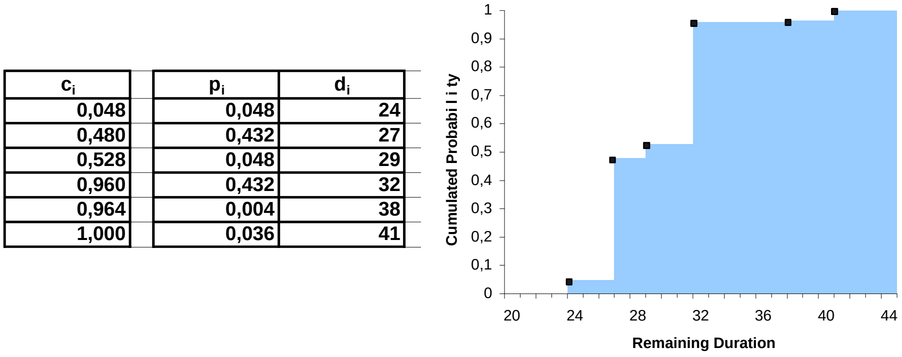
3. Duration histograms

For the representation of the probabilistic values for the duration of workflows, and the duration of (complex) activities due to conditional branches we propose *duration histograms.* Table 1 contains the execution durations d_i and the execution probabilities p_i for all 8 possible routes between E and T. The execution durations d_i of each path can be calculated by adding all execution durations of the activities which lie on the corresponding route.

Since we are typically not interested in all individual paths we aggregate this information deriving a table with all the different duration values and the cumulated probabilities (i.e. sum of the probability values of all paths with the same duration) Figure 2 shows the resulting table and its representation as histogram, thus this matrix is called *Duration*

Route	p_i	d_i
`EILNRST`	0.048	24
`EHLNRST`	0.432	27
`EILORST`	0.048	29
`EHLORST`	0.432	32
`EILPRST`	0.003	38
`EILQRST`	0.001	38
`EHLPRST`	0.027	41
`EHLQRST`	0.009	41

Table 1. *Possible routes between E and T (ordered by duration)*



c_i	p_i	d_i
0,048	0,048	24
0,480	0,432	27
0,528	0,048	29
0,960	0,432	32
0,964	0,004	38
1,000	0,036	41

Figure 2. Duration histogram subworkflow E to T

Histogram. Every matrix row describes one corner of a histogram, where the y-axis value is given by the cumulated branching probability c_i and the x-axis value by the duration d_i.

Formally we define duration histograms as follows:

Definition 1 (Duration Histogram) *A duration histogram D is a binary relation with n rows (p, d), (probability p and duration d).*

A duration histogram D is valid, if $\sum_{i=1}^{n} p_i = 1$ for $(p_i, d_i) \in D$.

An extended duration histogram D is a relation of n rows (p_i, c_i, d_i), (probability p, cumulated probability c, and duration d), with $\sum_{i=1}^{n} p_i = 1$, and $c_i = \sum_{d_j \leq d_i} p_j$ for $1 \leq i \leq n$.

A cumulated duration histogram is the projection of an extended duration histogram on the cumulated probabilities and the duration.

The duration histogram D of a node v (elementary activity, complex activity, workflow) is represented by a $(n \times 2)$-matrix, where each row holds one tuple (p_i, d_i); p_i represents a probability and d_i holds the associated remaining-execution-duration (that is the execution-duration between this activity and the final activity of the structure). The matrix is ordered by the duration d_i. In the remainder of this paper we use the probabilities or cumulated probabilities, whichever is more convenient in the situation since we can easily compute one from the other.

On (extended) duration histograms we define two basic operations: duration selection and probability selection.

Definition 2 (Selection) *For the cumulated duration histogram D, the duration selection operation is defined as:*
$\sigma^d_q D = min\{x \mid \exists (c, x) \in D : c \geq q\}$,
and the probability selection operation is defined as:
$\sigma^p_d D = min\{c \mid \exists (c, x) \in D : x \geq d\}$

Examples: For the duration histogram E of Figure 2: $\sigma^d_{0.8} E$ returns 32, the duration within which 80% of the workflows terminate. $\sigma^p_{28} E$ returns 0.528, the probability that the workflow terminates within 28 time units.

4. Calculation of duration histograms for workflows

We apply duration histograms to represent the temporal properties of a workflow in form of the *Probabilistic Timed Graph*, where each node of the workflow graph is adorned with a duration histogram representing the remaining time of the workflow. In this section we show how to calculate this graph depending on different workflow control-structures (sequence, conditional execution, parallel execution, loops) and introduce the required operations on workflow histograms (addition, disjunction, conjunction). Furthermore, we discuss the possible compression of duration histograms.

The Probabilistic Timed Graph will be calculated beginning with the final activity in reverse order of the dependencies of the workflow activities, whereby the calculation of each duration histogram needs its successors matrices. The final activity is initialized with a duration histogram with a single row $(1, 1, d)$, where d is the duration of the final activity.

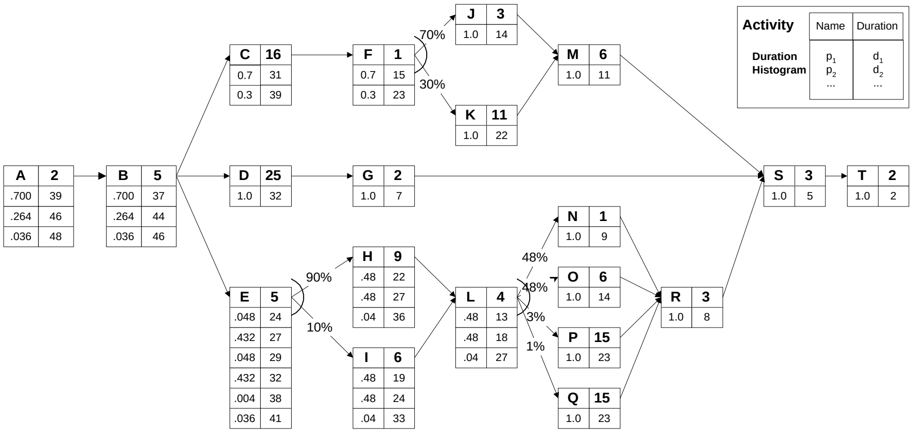


Figure 3. Example workflow with duration histograms calculated

4.1 Addition of a scalar for sequences

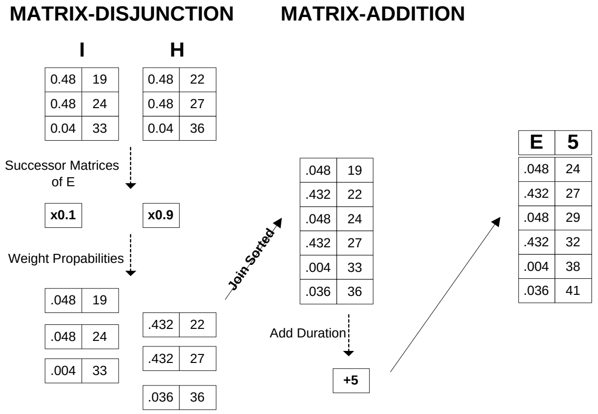
For two activities $R \rightarrow S$ connected by a sequence we calculate the duration histogram of activity R by adding it's duration to the duration histogram of S. Examples are the activities J, M, P or S (see also figure 3).

Definition 3 (Addition of scalar) *The addition of a scalar value c to a duration histogram D is defined as* $D + c = \{(p, d + c) \mid (p, d) \in D\}$

4.2 Adding duration histograms

There are two possibilities where adding a scalar value for expressing sequences of activities is not enough. First, it might be a sequence of composite (or complex) activities, resp. subworkflows, whose durations are expressed by duration histograms. Second we can use duration histograms also to express statistically distributed durations of elementary activities, too. In both cases, it is necessary to calculate the duration of the sequence of two activities by adding their duration histograms.

Definition 4 (Addition) *For the duration histograms B and C,* $B + C = \{(\sum p, d) \mid \exists (p_b, d_b) \in B, \exists (p_c, d_c) \in C, d = d_b + d_c, p = p_b * p_c\}$.



I	
0.48	19
0.48	24
0.04	33

H	
0.48	22
0.48	27
0.04	36

.048	19
.048	24
.004	33

.432	22
.432	27
.036	36

.048	19
.432	22
.048	24
.432	27
.004	33
.036	36

E	5
.048	24
.432	27
.048	29
.432	32
.004	38
.036	41

Figure 4. Conditional structure: matrix-disjunction and matrix-addition

4.3 Disjunction of duration histograms for conditional execution

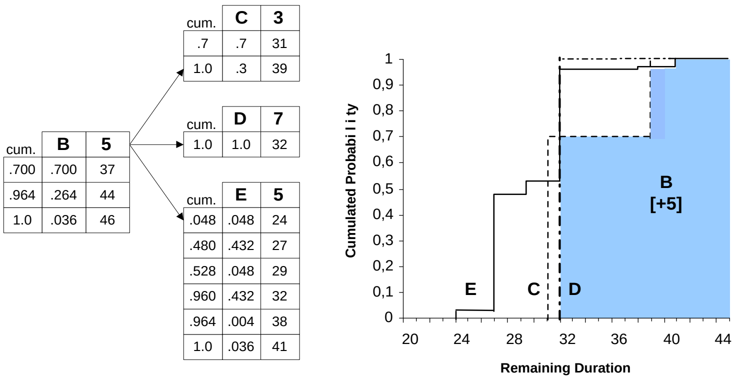
To calculate duration histograms for conditional structures, at first all successors matrices have to be *aggregated* into one result matrix, considering the given branching probabilities. For an activity A followed by an or-split with activity B with branching probability p and C with probability q the duration histogram is computed by a weighted disjunction of the matrices for B and C and addition of the duration of A.

The disjunction is computed by multiplying the probabilities in the duration histograms with their branching probabilities and merging these matrices.

Definition 5 (Disjunction) *For the duration histograms B, C and the branching probabilities q and $1-q$ the disjunction $(q, B) \vee (1-q, C) =$ $\{(p_i, d_i) \mid \exists p : (p, d_i) \in B \vee (p, d_i) \in C, p_i = q * \varsigma(d_i, B) + (1-q) * \varsigma(d_i, C)\}$, where $\varsigma(d, X) = p$, if $(p, d) \in X$, or 0, otherwise.*

It is easy to see that the sum of all probabilities of A is 1, and A is a valid duration histogram. The definition is easily extended to sets of successors.

For our running example (figure 3) the duration histogram of E is computed by $E = E.d + ((0.1, I) \vee (0.9, H))$ (see figure 4).



cum.	B	5
.700	.700	37
.964	.264	44
1.0	.036	46

cum.	C	3
.7	.7	31
1.0	.3	39

cum.	D	7
1.0	1.0	32

cum.	E	5
.048	.048	24
.480	.432	27
.528	.048	29
.960	.432	32
.964	.004	38
1.0	.036	41

Figure 5. Parallel execution: histogram superposition

4.4 Conjunction for parallel execution

After and-splits all succeeding routes will be processed concurrently. This means that the durations have to be aggregated, wherby the longest route determines the duration of the whole structure. Figure 5 visualizes this aspect. On the right side the duration histograms of C, D and E are superpositioned.

Being 'cautios', only maximum remaining-durations have to be considered – but now based on every possible probability. Thus the resulting matrix describes a histogram, which is composed by the right outer line of the superpositioned histograms of the successor-matrices. That means, when intersecting the y-axis parallel to the x-axis on every possible y-value, the remaining-duration which is farthermost right must be chosen. The grey hatched area in figure 5 visualizes the resulting matrix (which does not include the addition of Bs duration $d = 5$). This operation is called *Conjunction*: $B = \bigwedge(C, D, E)$. It is obvious that the resulting duration histogram does no longer hold values for each possible route between the activity and the end of the workflow. What it holds is the compressed time- and probability-information that the workflow-scheduler needs to make decisions (see also section 6).

Definition 6 (Conjunction) *Let B and C be cumulated duration histograms.* $B \wedge C = \{(q, max(\sigma^d_q B, \sigma^d_q C) \mid \exists y\ (q, y) \in B \cup C\}$

Again, this definition can easily be extended to conjunction of sets of duration histograms.

4.5 Iterations

For loops it might not be possible to exactly predict how many iterations will be executed during run-time. Therefore, we apply a similar approach to represent probabilities of iteration numbers and compute on this basis the duration of the whole loop structure.

4.5.1 Definition and iteration distribution. With regard to existing workflow-models the iteration structure is capsuled in a so called *Complex Activity*, that means it is embedded in an activity-structure, which in turn contains a workflow-graph itself and additionally the iterations probability information:

In analogy to duration histograms, the iteration histogram L of a complex activity is a binary relation of n rows (p, x) (probability p and iteration count x). Table 2 shows a possible iteration histogram L. This matrix can be produced empirically from statistical analysis of the workflow log.

Iteration x_i	Absolute Frequency	Relative Frequency p_i
1	2	0.0270
2	6	0.0811
3	9	0.1216
4	14	0.1892
5	15	0.2027
6	13	0.1757
7	7	0.0946
8	5	0.0676
9	3	0.0405
Sum	40	1.0000

Table 2. Distribution of iterations

Again p_i can be cumulated in increasing order and the result can be interpreted as follows:

- There is a 2.7% chance that the iteration will be executed once.
- There is a 10.81% (=2.7%+8.11%) chance that the iteration will be executed two times or less.
- ...
- There is a 100% chance that the iteration will be executed nine times or less.

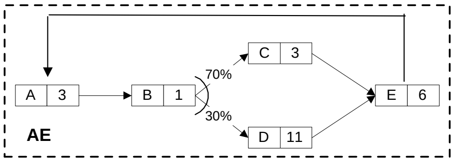


Figure 6. Iteration AJ embedded in complex activity

For the following we assume that the workflow-graph of figure 6 is embedded in the complex iteration activity AE. Furthermore, we assume that its iteration probability distribution L is given in table 2.

4.5.2 Splitting the Structure. First the duration histogram of AE must be calculated, according to the explanations of the previous sections:

p_i	d_i
0.7	13
0.3	21

Table 3. Initial duration histogram of AE

Table 3 represents the initial duration histogram (after one loop). Now the rest of the iterations, as given in the distribution, must be considered. For this reason it is necessary to split the iteration into a structure that is composed of sequences and conditionals which connect the complex activity AE multiple times, as shown in figure 7. The probabilities labeled on the edges can be taken from the iteration distribution L (see table 2).

In section 4.1 we introduced the matrix addition to calculate duration histograms considering two or more sequentially arranged activities. In the case of the splitted graph this technique has to be applied repeatedly. This operation is called *Matrix Multiplication.*

Definition 7 (Multiplication) *The multiplication of a duration histogram D with a cardinal number n is is defined as $n \cdot D = \sum_1^n D = D + D + \cdots + D$*

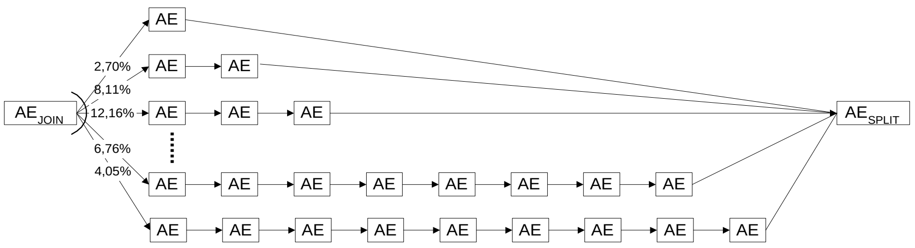


Figure 7. Iteration split into sequences and conditionals

After the multiplication a matrix-disjunction has to be performed on the resulting histograms to consider the branching probabilities (see table 4): $AE.D_{act} = ((0.027, D) \vee (0.0811, 2 \cdot D) \vee \cdots \vee (0.0405, 9 \cdot D)$.

Multiplication Result	x_i	p_i
$1 \cdot D = D$	1	0.0270
$2 \cdot D = 1 \cdot D + D$	2	0.0811
$3 \cdot D = 2 \cdot D + D$	3	0.1216
$4 \cdot D = 3 \cdot D + D$	4	0.1892
...		

Table 4. Matrix multiplication

5. Compression of duration histograms

With the algorithms presented so far it is possible that huge duration histograms might be created. To avoid too large histograms we introduce compression operations.

Duration histograms are compacted without loss of information resp. accuracy, by aggregating the duration histogram such that all durations are unique and the probabilities are summed up. Likewise, in the cumulated duration histograms, a row is deleted, if there is another row, with the same duration but smaller cumulated probability.

On compacted duration histograms we apply several compression techniques for decreasing the size of duration histograms. First we describe a method, where the loss of accuracy is minimized, then we present a method for pruning a duration histogram at predefined steps.

The goal of the technique is to keep the number of entries in the duration histogram on a defined level, without loosing to much informa-

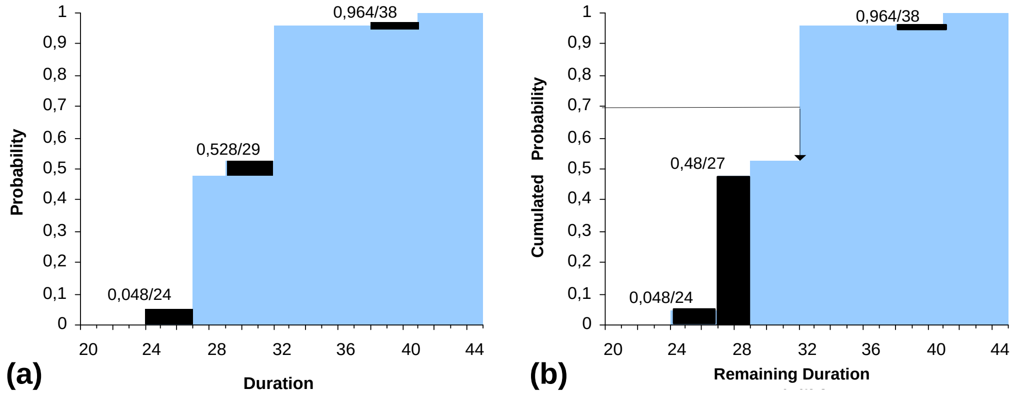


Figure 8. Compression of duration histogram

tion. This is achieved by iteratively searching for the row with the least information content in the histogram and removing this row.

Definition 8 (Operation: compression) *To compress a duration histogram D with $n = |D|$ entries (c_i, d_i), where $0 < i \leq n$, to size m, where $1 < m < n$, apply the following*

- *Add row (0.0) to D ($c_0 = 0$ and $d_0 = 0$)*
- *While $|D| > m$*
  ◦ *$area(j) = [(c_j - c_{j-1} * (d_{j+1} - d_j)]$ where $0 < j \leq n$*
  ◦ *Remove (c_j, d_j) where $\forall i \neq j : area(j) < area(i)$*

In the example three areas have been removed to reach the defined number of three entries in the duration histogram.

Figure 8 (b) illustrates a second compression approach, which is based on the idea, that entries below a specific threshold are not of interest for a workflow-scheduler and can therefore be skipped. In the example this threshold is 70% and since the duration histogram holds no row with $c_i = 0.7$ the next-lower entry $(0.528, 29)$ must be preserved. But the two areas below can be omitted. Additionally, the maximum number of entries for the final matrix was set to 3. Thus, the type(a)-matrix-compression has to be applied until this number is reached.

6. Application

In the previous sections we described how to calculate the timed probability graph of a workflow-structure containing sequences, conditional executions, parallel executions or loops. The result is a duration histogram for every activity, representing the duration distribution of the rest of the workflow. This information can be used to calculate the probability of meeting defined deadlines.

6.1 Build Time

For all activities we can calculate internal deadlines. There are different possibilities how this can be achieved. The easiest way is to determine a deadline for the entire workflow. For our example we suggest a deadline $DL = 48$, because this is the highest remaining-duration value of the start-activity A (see figure 3). Afterwards every duration value d_i in each duration histogram can be used to calculate a starting-time value $t_i^{start} = DL - d_i$. Additionally, the ending-time $t_i^{end} = t_i^{start} + \partial$, can be determined for each activity, where ∂ is the duration of the actual activity. Table 5 shows this for activity E.

p_i	c_i	d_i	$t_i^{start} = DL - d_i$	$t_i^{end} = t_i^{start} + \partial$
0.048	0.048	24	24	29
0.432	0.480	27	21	26
0.048	0.528	29	19	24
0.432	0.960	32	16	21
0.004	0.964	38	10	15
0.036	1.000	41	7	12

Table 5. Starting-times and finishing-times of activity E

Amongst other things, the following conclusions can be drawn: The earliest point in time, when E is able to finish is 12. The latest point in time, when E is allowed to finish, without risking the workflows deadline, is 29. Should E finish within 29 TUs, there is still a 4,8% chance that the workflow-instance finishes within the given deadline of $DL = 48$.

These calculations enable us to use the timed probability graph to solve typical build-time problems, like the above mentioned deadline estimation and validation.

6.2 Run Time

With these calculated values t^{start} or t^{end} we implement simple but effective escalation-warning mechanisms, for example the traffic-light model: During build-time two values are defined. The first determines the workflows state-change from green to yellow (warn) and the second determines the state-change from yellow to red (alarm). As long as the workflow is *green* everything is ok, if the state changes something has to happen. According to the new state different escalation actions can be invoked, like skipping unnecessary (optional) tasks, calling for administrators help or even termination of the workflow-instance.

For an example, let's set the first threshold-value is 90% and the second threshold-value is 50%. Assuming the workflow-instance just finished activity E requiring 22 TUs so far, the workflow-status has to be changed from green to yellow, because the 90% threshold has not been reached. This can be determined by comparing the current time and the threshold-value with the matrix of E. If no exact cumulated probability value can be found, the nearest higher value must be chosen, in this case 96%. The corresponding t^{end}-value is 21, which is clearly below 22, thus the state has to be changed.

The thresholds can be set freely, from risky to conservative adjustments. Of course, it is also possible to employ a more fine grained escalation scheme with more intermediate threshold values. The important contribution of our approach is that we can now define these threshold values in terms of probabilities of deadline violations.

7. Conclusions

It is imperative that current and future workflow management systems provide the necessary information about a process, its time restrictions, and its actual time requirements to process modelers and managers. At build-time, when workflow schemas are defined and developed, workflow modelers need means to represent time-related aspects of business processes (activity durations, time constraints between activities, *etc.*) and check their feasibility. At run-time, when workflow instances are instantiated and their executions are started, process managers should be able to adjust time plans (*e.g.,* extend deadlines) according to time constraints and any unexpected delays. Furthermore, they need pro-active mechanisms for being notified about possible time constraint violations so that they can take the necessary steps to avoid time failures. In this paper we introduced a new method for representing different durations together with the probabilities for different durations of workflow executions in form of duration histograms. The operations we defined on these duration histograms allow the calculation of time plans for workflow execution and to reason about the probability of deadline violations. This should bring better scheduling decisions and improved escalation strategies for workflow execution.

References

[1] C. Bussler. Workflow Instance Scheduling with Project Management Tools. In *9th Workshop DEXA'98*, 1998. IEEE Computer Society Press.

[2] P. Dadam, M. Reichert, and K. Kuhn. Clinical workflows - the killer application for process-oriented information systems? In *4th International Conference on Business Information System (BIS 2000)*, pages 36–59, Poznan, Poland, 2000.

[3] J. Eder, E. Panagos, and M. Rabinovich. Time constraints in workflow systems. In *Proc. International Conference CAiSE'99*. Springer Verlag, 1999.

[4] D. Georgakopoulos, M. Hornick, and A. Sheth. An overview of workflow management: From process modeling to workflow automation infrastructure. *Distributed and Parallel Databases*, 3(2):119–153, 1995.

[5] D. Hollingsworth. The workflow reference model. Draft 1.1 TC00-1003, Workflow Management Coalition, July 1995.

[6] B. Kao and H. Garcia-Molina. Deadline assignment in a distributed soft real-time system. In *Proceedings of the 13th International Conference on Distributed Computing Systems*, pages 428–437, 1993.

[7] B. Kao and H. Garcia-Molina. Subtask deadline assignment for complex distributed soft real-time tasks. Techn. Report 93-1491, Stanford University, 1993.

[8] P. Lawrence. *Workflow Handbook 1997.* John Wiley & Sons, 1997.

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

[10] E. Panagos and M. Rabinovich. Escalations in workflow management systems. In *DART Workshop*, Rockville, Maryland, November 1996.

[11] E. Panagos and M. Rabinovich. Predictive workflow management. In *Proceedings of the 3rd International Workshop on Next Generation Information Technologies and Systems*, Neve Ilan, ISRAEL, June 1997.

[12] E. Panagos and M. Rabinovich. Reducing escalation-related costs in WFMSs. In *NATO Advanced Study Institue on Workflow Management Systems and Interoperability*, Istanbul, Turkey, August 1997.

[13] H. Pozewaunig, J. Eder, and W. Liebhart. ePERT: Extending PERT for workflow management systems. In *First European Symposium in Advances in Databases and Information Systems (ADBIS)*, St. Petersburg, Russis, 1997.

[14] Workflow Management Coalition, Brussels, Belgium. *Glossary: A Workflow Management Coalition Specification*, November 1994.