Advances in Agile Manufacturing – ICAM 2003
©2003 China Machine Press, ISBN 7–111–13395 –1

Time Constraints in Workflow Models

Weiping Li, Yushun Fan
National CIMS Engineering Research Center,
Department of Automation, Tsinghua University, Beijing, China, 100084.
liweipingmail@yahoo.com.cn, fan@cims.tsinghua.edu.cn

ABSTRACT: Time management is an important function of workflow management systems. The corresponding aspects include calculating the process execution time, identifying the critical path of a workflow process, avoiding the violation of time constraints such as deadline violations, time distance violations, satisfying Fixed Date Constraints and time difference constraints. Six time constraints are given in this paper and the method for identify the critical path of a workflow process is given accordingly. With the time constraints the workflow designers, executors and managers can do better in the workflow modeling and workflow running.

Key Words: workflow model, time constraint, directed graph and critical path

1 INTRODUCTION

A workflow is the automation of a business process, in whole or part, during which documents, information or tasks are passed from one participant to another for action, according to a set of procedural rules [1]. The automation of a business process is described within a workflow definition, which identifies the various workflow activities, dependence relationship and associated control data used to manage the workflow. An activity is naturally defined task in a workflow that forms one logical step within a process. A dependence relationship in traditional workflow processes can be of two types: execution dependence for control flow and information dependence for information flow [2]. When modeling an activity the role, input/output data and time duration are designated to describe a real process. While the relationship between activities are also designated which responsible for the sequence when the process is executing.

A workflow management system is a system that completely defines, executes and manages the workflow through the execution of software whose execution order is driven by a computer representation of the workflow logic. A workflow management system usually consists of three parts: (1) the build-time functions, which are responsible for defining and managing the workflow processes; (2) the run-time functions, which are responsible for executing, monitoring and managing the workflow processes in an operational environment; and (3) the run time interactions with human users and application mechanisms for processing the activities. A workflow management system consists of software components to store and interpret workflow definitions, create and manage workflow instances and control their interaction with workflow participants and applications. A workflow instance is the representation of a single enactment of a process. It is created and managed by a workflow management system [2].

The time constraints and time management are very important in the designing and managing of workflow model. But the current commercial products offer very limited function of time representation and management [3]. The researchers believe that time management should be part of the core management functionality provided by workflow systems to control the lifecycle of processes [3,4,5]. Time difference is introduced into workflow model [6]. Time Bound between events and Fixed-Date Constraint are defined in [7]. Workflow time management issues such as computing activity deadlines, checking time constraints, and identifying the critical path are mentioned in Reference [8].

Many constraints are discussed in different point of view. But there doesn't exist a perfect model that can take into account these constraints together. Thus six time constraints are provided in this paper. These time constraints work together to ensure that the business process can be modeled correctly and that the corresponding workflow instances of the business process can be executed properly in the-workflow systems. The algorithm of calculating the critical path of a certain workflow is given. The calculate result can help the workflow executor to pre-

dominate the running status of a workflow process.
The remainder of the paper is organized as follows. Section 2 describes a workflow model which time constraints are detailedly defined. Section 3 given the workflow control routing and some special control condition is discussed. Section 4 describes how to calculate the critical path of a workflow with the 6 time constraints in the workflow build time and run time. Finally we conclude the paper with the contribution and further work in section 5.

2 TIME CONSTRAINTS IN WORKFLOW MODEL

2.1 Workflow model

A workflow is a collection of activities and the dependencies between activities. Activities correspond to individual tasks in a business process. Dependencies determine the execution sequence of activities and the data flow between these activities. An executor is responsible for the execution of an activity and some input/output date and resources are necessary for the activity. When modeling a business process all the factors have to be taken into account.
Many works for workflow modeling are based on the Petri Nets approach [9]. The XML based method, i.e. WPDL, is also given by workflow management coalition [10]. In this paper a workflow process is modeled by the directed graph in which nodes correspond to activities and edges correspond to dependencies between them. A process is denoted by a directed graph,

$$G = < N, C >, \tag{1}$$

Where *N* is the set of all the activities, *C* is the edges in the graph *G*. An node *N*, which correspond to a activity, is denoted by

$$N = < D, E, R, T >, \tag{2}$$

Where *D* is the input/output data that an activity will use, *E* is the executor or the software component of an activity, *R* is the resource related to the activity and *T* is the time constraints. Here we just focus on the time constraints.

2.2 Time constraints

The time constraints of an activity is

T=< Time Duration, Force Start Time, Deadline, Fixed Date constraint, Time Difference, Time Distance >, (3)

Where,
Time Duration: The duration assigned to an activity. Time Duration is necessary for every activity in a workflow process. It is assigned by the workflow designer according to the experience value.
Force Start Time: Force Start Time is a certain time that requires a certain activity can only be executed after it.
Deadline: Deadline is a time based scheduling constraint which requires that a certain activity be completed by a certain time (the "deadline") [1].
Fixed Date constraint: An activity can only be executed on certain fixed dates. The Fixed Date maybe *on Monday,* the *last day of every month or even 1 to 3 every month*. An example is that sanction of a certain file can be done on every Monday. Then *'Monday'* is the fixed date. The Fixed Date constraint can be denoted as,
fd_B, The left bound of the constraint, i.e. the begin time that an activity can start.
fd_E, The right bound of the constraint, i.e. the end time that an activity have to be finished otherwise it have to wait until next fixed date.
Time difference: The time difference between two activities is caused by different time zones. To meet the requirement of global business some workflow should be distributed in geography. Thus different activities of one business process have to be executed in different time zones.
Time Distance: The time duration between activity A1 and A2, where A1 is the predecessor of A2. Sometimes the successor of an activity must be waiting for a certain time because some essential reasons such as law, business rules. And sometimes a certain time distance is required to wait that the data and files are transferred from one person to another who is in charge of the successor activity.
For convenience the 6-tuple time constraints can be write as follows.

$$T = < t, t_{fs}, dl, fd, t_{dif}, t_{dis} > \tag{4}$$

The value of each time constraint, if have, is assigned by the workflow designer according to the experience value in the workflow build time. For the sake of simplicity we assume that all the 6 time constraints are deterministic. The time duration, time distance and time difference are expressed in some basic time units and in integer value.
With the 6-tuple time constraints the time attribute of an activity can be described explicitly. Except time duration not all the other 5 time constraints is compulsory. Any of them can be assigned to an activity or not according to the truth.
Besides these time constraints there are some time constraints follow implicitly from control dependencies and activity durations of a workflow model. They arise from the fact that an activity can only start when its predecessor activities have finished. Such constraint is called the structural time constrains since they abide the control structure of the workflow.

3 WORKFLOW CONTROL ROUTING

3.1 Control routing

In a workflow instance that managed by a workflow

systems the activities is executed according to the workflow model. There exist some control routing in the workflow model that formed by the dependencies between activities. The activities may be executed in sequence or parallel. Sequential routing is that a segment of a process instance under enactment by a workflow management system, in which several activities are executed in sequence under a single thread of execution. Parallel routing is that a segment of a process instance under enactment by a workflow management system, where two or more activity instances are executing in parallel within the workflow, giving rise to multiple threads of control.
The workflow control routings are determined by the workflow control condition, e.g. Sequence, AND-Split, AND-Join, OR-Split, OR-Join, and LOOP.
The graphical expressions of them are shown in Figure 1～6. As described in Ref. [1], these control conditions are defined as follows.
Sequence: A segment of a process instance under enactment by a workflow management system, in which several activities are executed in sequence under a single thread of execution.
AND-Split: A point within the workflow where a single thread of control splits into two or more threads which are executed in parallel within the workflow, allowing multiple activities to be executed simultaneously.
AND-Join: A point in the workflow where two or more parallel executing activities converge into a single common thread of control.
OR-Split: A point within the workflow where a single thread of control makes a decision upon which branch to take when encountered with multiple alternative workflow branches.
OR-Join: A point within the workflow where two or more alternative activity(s) workflow branches re-converge to a single common activity as the next step within the workflow.
Iteration: A workflow activity cycle involving the repetitive execution of one (or more) workflow activity(s) until a condition is met.
In the parallel routing, an AND-Split usually combines with an AND-Join and an OR-Split with an OR-Join [11]. But also an AND-Split can combines with an OR-Join, for example, three bidders do the same prepare work in parallel (AND-Split) and the tenderee determines which one is selected (OR-Join), which forms the (AND-Split, OR-Join) couple.

3.2 The time duration of iteration

When calculate the time attributes of an activity the time duration of Iteration must be considered in advance. In a typical Iteration structure, see Fig. 6, a probability should be given for the need of redo the activities involved. Let the probability of reiteration is p, then the total time duration of the Iteration block can be calculated as follows.
The execution time of the iteration can be estimated as the average execution time of activity A2. When calculate the time of the process the execution time of the iteration can be replaced by the average time. Let's the time duration of activities are, t_{A2} and t_{A4}, then, the total average execution time T can be,

$$T = t_{A2} + p(t_{A2} + t_{A4}), \tag{5}$$

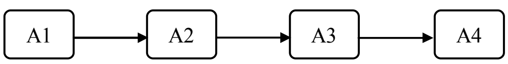

Figure 1. Sequence

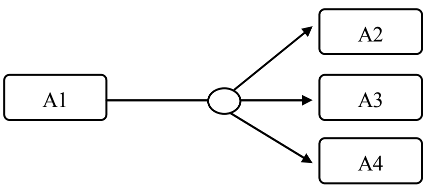

Figure 2. AND-S.plit

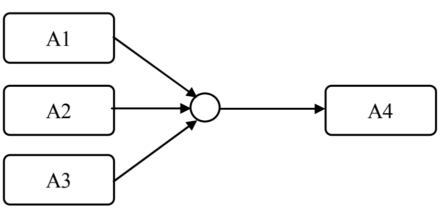

Figure 3. AND-Join

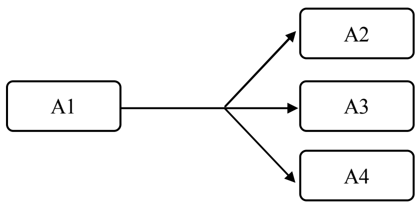

Figure 4. OR-Split

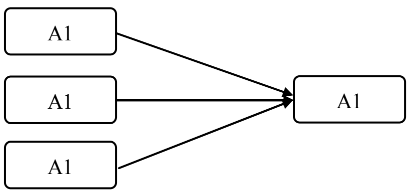

Figure 5. OR-Join

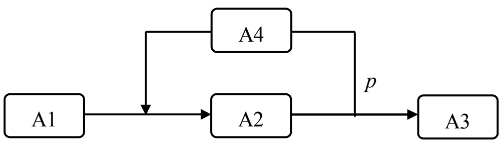

Figure 6. Iteration

i.e.,

$$T = t_{A2}(1+p) + pt_{A4} \tag{6}$$

4 WORKFLOW CRITICAL PATH

With the workflow model and the time constraints the critical path can be identified while the start and finish times of activities can also be calculated. After identifying the critical path of a workflow process we can benefit from it. The workflow model designer, the workflow executor and the workflow manager all need the results to instruct their work.
When modeling the workflow process designers should assign the time constraints for every activity, i.e. Time Duration, Force Start Time, Deadline, Fixed Date constraint, Time Difference, and Time Distance. The general time information on the workflow model and activities is calculated based on the 6-tuple time constraints and the process structure. The workflow designer will find out the temporal bottlenecks and will try to optimize the process according to these information.
When executing the model the time information is calculated in real time and the whole execute process is monitored. The time information can be used to adjust the process if any time constraints is violated. The workflow executor will schedule the tasks to avoid any time constraint violations in advance.
When computing the critical path of a given workflow process the following relative time information of an activity is given: E_S, E_F, L_S and L_F. Where,

E_S stands for the earliest start time of an activity,
E_F stands for the earliest finish time of an activity,
L_S stands for the latest start time of an activity and
L_F stands for the latest finish time of an activity.

Depending on the control dependencies between activities in a workflow process, the E_S, E_F, L_S and L_F are computed in this way. First, a forward traversal of the workflow process is required for the calculating E_S and E_F. At the beginning of this traversal E_S and E_F of all activities without predecessors are set to 0(build time) or the start time of the process (running time). Second, a backward traversal of the workflow process is required for the calculating L_S and L_F. At the beginning of this traversal L_S and L_F of all activities without successors are set to their corresponding E-values (if no deadlines are set to them) or their deadlines.
Assume A and B are two adjacent activities, i.e. A is the predecessor of B. Some rules needed when computing the critical path are follows.

- The E_S of B is the E_S of A plus the time duration of A,

$$E_{SB} = E_{SA} + t_A . \tag{7}$$

- If some time distance required, the E_S of B is

$$E_{SB} = E_{SA} + t_A + t_{disA} . \tag{8}$$

- If two adjacent activities are in different time zones, the time difference must be take into account. If the activity B can be executed exactly after the activity A is finished, i.e. the executor in time zone B is in working hours, then the time difference can be neglected. Otherwise some waiting time is need until the executor in the time zone B can begin to work, i.e. in working hours.

Time zone B is in the working hours,

$$E_{SB} = E_{SA} + t_A , \tag{9}$$

or, time zone B isn't in working hours.

$$E_{SB} = E_{SA} + t_A + t_{wd} , \tag{10}$$

When calculating the time information, we have to convert the time in time zone B into time zone A to get the waiting time. Here we use the time in time zone A to represent the time in time zone B,

$$t_{wd} = t_{difB} - t_{difA} + t_{interval} , \tag{11}$$

where $t_{interval}$ is the time that we have to wait form now until new working hours in time zone B, which will be determined in the run time.

- If there is a fixed data constraint on activity B, two situations must be taking into account. If the time $E_{SA} + t_A$ is located in the space of the fixed date of activity B, i.e.,
  $fd_B < E_{SA} + t_A < fd_E$,

Then

$$E_{SB} = E_{SA} + t_A , \tag{12}$$

Else, $E_{SA} + t_A < fd_B$, or, $E_{SA} + t_A > fd_E$,

$$E_{SB} = E_{SA} + t_A + t_{wf} . \tag{13}$$

t_{wf} is the wait time that activity B has to wait form now until fd_B.

The t_{wf} of a fixed time constraints can be replaced by the average time, see Fig. 7.

$d_1 = fd_E - fd_B$ is the time duration of the fixed date constraint,

$d_2 = fd_{B2} - fd_{E1}$ is the cycle of the fixed date constraint.

When an activity is finished the probability that the E_F located in the area of fixed date and the area of

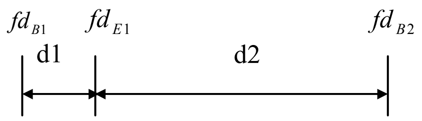

Figure 7. The fixed date constraint.

interval is the same. Then the probability density is,

$$f(x) = 1/(d_1 + d_2), \tag{14}$$

and the waiting time is,

$$t(x) = \begin{cases} 0, x \in [0, d_1) \\ d_1 + d_2 - x, x \in [d_1, d_1 + d_2) \end{cases} \tag{15}$$

then,

$$t_{wf} = \int_0^{d_1+d_2} t(x) f(x) dx = \int_{d_1}^{d_1+d_2} (d_1 + d_2 - x) \, 1/_{d_1+d_2} dx$$

$$t_{wf} = d_2^2 /_{2(d_1+d_2)} \tag{16}$$

4.1 Build time calculations

In the build time, the time constraints of every activity in all workflow models are given according to the experience or estimated value. These time information can be used to compute the critical path of the workflow process and some basic time information such as E_S, E_F, L_S and L_F. Because the workflow model is static, which is not running, the force start time, deadline and time difference can't be identified in advance. Thus these three time constraints won't be take into account. Although there exist OR-Join that caused by the AND-Split or OR-Split we treat them just like they are AND relationship, because one can't make the decision until the run time.
The build time calculations algorithm is as follows.

1 Forward calculation:
For all activities in a workflow process
If activity A is start node, let

$$E_{SA} = 0, E_{FA} = t_A$$

If activity B isn't the start node, let

$$E_{SB=} Max\{E_{FA} + \max(t_{wfB}, t_{disA}) \mid A \in B.pred\}$$

$$E_{FB=} E_{SB} + t_B .$$

End for

2 Backward calculations:
For all activities in a workflow process
If activity A is the end node, let

$$L_{FA} = E_{FA}, L_{SA} = L_{FA} - t_A$$

If activity B isn't the end node, let

$$L_{FB} = Min\{L_{SA} - \max(t_{wfA}, t_{disB}) \mid A \in B.succ\},$$

$$L_{SB} = L_{FB} - t_B$$

End for

4.2 Runtime calculations

After the instantiation of a workflow process the start time of a process and the start activity was identified. The start time is a real time and the time information of all the other activities can also be identified according to this time. When calculating the critical path all the six time constraints should be take into account.
Assume the start time is T. The run time calculations algorithm is as follows.

1 Forward calculation:
Before calculate the critical path of an running workflow process, Check and find out the OR-Split & OR-Join pair, if the OR-Split condition is given then cut the corresponding branch, a new directed graph is given. The calculation is performed on the 'new' process.
For all activities in a workflow process
If activity A is start node, let

$$E_{SA} = T, E_{FA} = t_A$$

If activity B isn't the start node, let

$$E_{SB=} Max\{(E_{FA} + \max(t_{disA}, t_{wdB}, t_{wfB})), t_{fsA} \mid A \in B.pred\}$$

$$E_{FB=} E_{SB} + t_B .$$

End for

2 Backward calculations:
For all activities in a workflow process
If activity A is the end node, let

$$L_{FA} = Min\{E_{FA}, dl_A\}, L_{SA} = L_{FA} - t_A$$

If activity B isn't the end node, let

$$L_{FB} = Min\{(L_{SA} - \max(t_{wdA}, t_{wfA}, t_{disB})), dl_B \mid A \in B.succ\}$$

$$L_{SB} = L_{FB} - t_B$$

End for

4.3 Time constraints violation

Since there are so many time constraints in one workflow model there may exist some violation of the time constraints when the workflow model is running. With the critical path algorithm the workflow executor can find any time constraints in time and adjust the execution of activities to conquer it.
Deadline violation is often occurs in the real business process. If the potential deadline violation will occur, there are three methods to overcome the diffi-

culty.

- Adjust the time durations of activities that are under scheduling to reduce the whole time duration of the whole activity block.
- Try to postpone the deadline of the very activity that will be violated.
- Try to find another path to replace the current path. This method requires the workflow model define some spare path in advance.

5 CONCLUSIONS

In order to model the business processes more accurately 6-tuple time constraints is defined in this paper. These time constraints work together to ensure that the business process can be modeled correctly and that the corresponding workflow instances of the business process can be executed properly in the workflow systems. The algorithm of identifying the critical path of a certain workflow process is given based on the 6 time constraints. Finding the critical path gives us important information of a workflow model. More than anything else, the critical path may help to find the workflow bottleneck points. Hence, high performance workflow systems can be achieved by efficiently managing the critical path.
Our future work focus on: establishing dynamic workflow model based on the time constraints; making a workflow management based on the dynamic workflow model, which will be built on the J2EE architecture; scheduling the execution of activities of different workflow process as a whole system in the run time.

ACKNOWLEDGEMENTS

The research was supported by the National Science Foundation of China (No. 60274046).

REFERENCES

[1] Workflow Management Coalition, 1999. Terminology and Glossary, Document Number WFMC-TC-1011, Issue 3.0.

[2] Lawrence, P. (Ed.), 1997. Workflow Handbook. Wiley, New York.

[3] Marjanovic, O., Orlowska, M.E.. 1999. On modeling and verification of temporal constraints in production workflows. Knowledge Information Systems. 1, 157-192.

[4] Li Hui-fang, fan Yushun, Managing 2002. Time in Workflow Systems-an OverView, Journal of Software, Vol. 13, No.4, 1-7.

[5] Eder J., Panagos E., Pozewaunig H., Rabinovich M.. 1999. Time Management in Workflow Systems. In: Abramowicz W, Orlowska M E, BIS'99 3rd International Conference on Business Information Systems, Heidelberg: Springer Verlag London Berlin, 265-280.

[6] Zhuge H., Cheung T., Pung H.. 2001. A Timed Workflow Process Model . The Journal of Systems and Software, Vol. 55, 231-243.

[7] Eder J., Panagos E., Rabinovich M.. 1999. Time Constraints in Workflow Systems. In: Proc. 11th Conference on Advanced Information Systems Engineering (CAISE'99), Heidelberg, Germany, 286 - 300.

[8] Jin Hyun Son, Myoung Ho Kim, 2001. Improving the performance of time-constrained workflow processing, The Journal of Systems and Software, Vol. 58, 211-219.

[9] Van der Aalst W.M.P.. 1998. The Application of Petri nets to Workflow Management. The Journal Of Circuits, Systems and Computers, 8(1), 21-66.

[10] Workflow Management Coalition, 2001. Workflow Process Definition Interface – XML Process Definition Language, Document Number WfMC-TC-1025.

[11] Duk-Ho Chang, Jin Hyun son, Myoung Ho Kim, 2002. Critical path identification in the context of a workflow, Information and software Technology, Vol. 44, 405-417.