ARTICLE IN PRESS

Information Systems ∎ (∎∎∎∎) ∎∎∎–∎∎∎

Contents lists available at ScienceDirect

Information Systems

journal homepage: www.elsevier.com/locate/infosys

Controlled violation of temporal process constraints – Models, algorithms and results

Akhil Kumar, Russell R. Barton

Smeal College of Business, Penn State University, University Park, PA 16802, USA

ARTICLE INFO

Article history:
Received 13 November 2015
Received in revised form
23 May 2016
Accepted 7 June 2016

Keywords:
Process management
Temporal workflow model
Constraint violation

ABSTRACT

While there has been much work on modeling and analysis of temporal constraints in workflows in the context of many real-world applications, there has not been much work on managing violations of temporal constraints. In real-time workflows, such as in medical processes and emergency situations, and also in logistics, finance and in other business processes with deadlines some violations are unavoidable. Here we introduce the notion of controlled violations as the ability to monitor a running process and develop an approach based on constraint satisfaction to determine the best schedule for its completion in a way so as to minimize the total penalty from the violations. The violations are evaluated in terms of metrics like number of violations, delay in process completion, and penalty of weighted violations. We also relate our work to the concept of controllability in literature and show how it can be checked using our method. Finally, we analyze the properties of our approach and also offer a proposal for implementation.

1. Introduction

Many real-world workflows run under time constraints. A mortgage application received by a bank from a customer for the purchase of a house must go through steps like credit check, property appraisal, title search, etc. within a fixed amount of time with clear deadlines for each stage. Similarly, a business order for a desktop computer must be assembled, packed, loaded and shipped according to a clear schedule. In a hospital setting a patient proceeds through steps like testing, diagnosis, pre-operation care, surgery, post-operation care, recovery and discharge as per prescribed guidelines. These are all examples of time-sensitive service processes where timeliness has a huge impact on service quality.

Medical processes are particularly sensitive to the observance of strict temporal guidelines for the success of a treatment procedure. Some examples of such guidelines that arise in a medical process (say for the treatment of a fracture) are:

- A radiologist's report must be submitted within 24 h of a CT scan.
- If surgery is needed it must take place within a week of the radiologist's report.
- Antibiotics must be taken for 3 days before surgery.
- A blood thinner like Aspirin must be stopped 24 h before surgery.
- The patient must recover in the hospital for 2 days before being discharged.
- The total time from patient admission to discharge should not exceed 7 days.

In modeling such time-aware processes [6,7,17,20,26], the duration of each activity (or task) is provided as a

E-mail addresses: akhil@psu.edu (A. Kumar), rbarton@psu.edu (R.R. Barton).

http://dx.doi.org/10.1016/j.is.2016.06.003
0306-4379/

range, or just a lower or upper limit. For example in a medical process the duration of the patient admission activity is, say, between 10 and 20 min. By associating such durations with each activity one can determine expected minimum and maximum times for each execution path of the workflow from start to end. Moreover, deviations from the expected times can be monitored, and appropriate messages and alerts can be generated to draw attention. Another aspect of temporal workflows relates to inter-activity constraints that impose restrictions on the elapsed time between one activity and another. Further they may be specified with reference to the start or finish time of the respective activities. A variety of temporal constraints can be imposed on a workflow [18]. While general types of semantic constraints have been studied in literature [15,23,24], there is less work on temporal constraints.

A temporal workflow should represent various temporal patterns and relationships among activities. Temporal patterns and ways of reasoning with them are discussed in [1,5,12]. To some extent, planning a temporal workflow is like scheduling with concepts like early (late) start times and finish times for various activities [10,11]. Another concept in the context of temporal workflows is the idea of controllability [7,17,22] which relates to the flexibility present in a workflow schedule. The work on controllability is based on the notion of conditional simple temporal networks [31] which were developed in the context of planning. A workflow that allows activity durations to fall anywhere within their allowed range and still complete successfully is said to be *dynamically controllable*. Algorithms for dynamic controllability are discussed in [7,17,22].

In this paper, we take the view that while on the one hand guidelines are very important, yet on the other it is not always possible to enforce them very strictly in practice. Hence, there must also be some leeway or allowance for deviations from the guidelines. Some unexpected delays may occur for various reasons at run time (e.g. patient admissions is backed up; CT machine has broken down, etc.) and lead to violations of constraints. If a task deviates slightly from its prescribed temporal range, it does not mean that the workflow cannot proceed. The natural question to pose then is: how will this deviation or violation affect the rest of the workflow? If the effect is small then the workflow can continue normally. Our goal in this paper is to develop a model that can take into account the possibility of violation of various constraints and explore the tradeoffs among the violations. Thus, if antibiotics medication has to be taken for three days before surgery and this will delay the surgery, there is a tradeoff between reducing the duration of the medication and delaying the surgery.

The novel aspect of our work is that we allow for constraints to be violated by introducing relaxation variables in our model, thus allowing for "graceful degradation." Our approach is based on constraint satisfaction with respect to an objective function. Each temporal constraint (both intra-activity and inter-activity) can be expressed as a linear equation(s). By checking if the constraints are consistent one can verify if they will all be satisfied. These variables assume values equal to the amount of violation in a constraint to force satisfaction. At the same time we also associate penalties with each violation, e.g. for every time unit of delay in start of surgery beyond the guidelines. Finally, these penalties are aggregated and minimized in an objective function.

This paper is a comprehensive extension of an earlier work [16] and includes detailed algorithms to describe our methodology, and extensive design- and run time analyses of temporal workflows. There is an expanded coverage of repetition structures like loops, formal results, claims of correctness and completeness related to our methodology, and a proposal for implementation.

This paper is organized as follows. In Section 2 we discuss a basic model for describing temporal constraints and show how it can be translated into structural and temporal constraint equations. Then, in Section 3, we describe how the approach was implemented and tested. Next, Section 4 extends our approach for managing violations of constraints and develops a formal optimization model based on penalties. Section 5 discusses how our approach can be extended to more complex control flow structures involving overlapping and repetitive activities. Later, Section 6 discusses some analytical results that highlight the features of our approach, while Section 7 gives an implementation proposal for our methodology. Finally, Section 8 discusses related work, and the last section gives the conclusions and shares some thoughts for future work.

2. Basic notation and modeling approach

2.1. A simple temporal model

To create a temporal model of a process two types of constraint models are combined: (1) basic structural constraint model, and (2) temporal constraint model. The structural constraints capture the control flow of the process to coordinate the proper sequence in which the tasks occur. The temporal flow model considers the permitted durations of each activity and the minimum or maximum gaps between them.

Def. 1. A general temporal process model TP can be represented as:

$TP = (\mathcal{T}, \mathcal{A}, \mathcal{X}, \mathcal{E}, \mathcal{TD}, \mathcal{TI})$
Where
$\mathcal{T}$: set of task nodes, T1, T2, …
$\mathcal{A}$: set of AND control nodes, A1, A2, …
$\mathcal{X}$: set of XOR control nodes, X1, X2, …
$\mathcal{E}$: set of edges among the nodes in {$\mathcal{T}$, $\mathcal{A}$, $\mathcal{X}$}
$\mathcal{TD}$: set of task duration ranges:{(Ti, Dimin, Dimax),…}, where Dimin, Dimax $\in \mathcal{R}+$
$\mathcal{TI}$: set of additional inter-task constraints: {(Ti, Tj, S|F, S|F, TIi_min, TIi_max), …}, TIi_min, TIi_max $\in \mathcal{R}+$

Fig. 1 shows an example of a simple temporal model. It shows the control flow, along with [min,max] durations of each task and inter-task constraints. It can be expressed as:
$\mathcal{T}$: {T1, T2, …, T6}
$\mathcal{A}$: { A1, A2}
$\mathcal{X}$: { X1, X2, X3, X4}

Please cite this article as: A. Kumar, R.R. Barton, Controlled violation of temporal process constraints – Models, algorithms and results, Information Systems (2016), http://dx.doi.org/10.1016/j.is.2016.06.003