Floating Task: Introducing and Simulating a Higher Degree of Uncertainty in Project Schedules

Sanja Lazarova-Molnar and Rabeb Mizouni
College of Information Technology
United Arab Emirates University
United Arab Emirates
{sanja, mizouni}@uaeu.ac.ae

Abstract

Despite several attempts to accurately predict duration and cost of projects, simulation models in use are still over simplified and nonrealistic. They often fail to cope with real-life scenarios and uncertainty.

In this paper we use the proxel-based simulation method to analyze and predict duration of project schedules exhibiting high uncertainty due to typical on-the-fly human decision behavior. The proxel-based simulation is an approximate simulation method that is more precise than discrete-event simulation.

To model uncertainty, we introduce a new type of task, state-dependent (floating) task that supports and demonstrates a high degree of uncertainty and depends on various parameters in the schedule. For example, the probability distribution of the duration of a task can change depending on the team that performs it. Thus, this kind of task can be used to model the frequent re-scheduling in a project. We use software development process to illustrate our approach.

Keywords: Project scheduling, simulation, uncertainty, human resource allocation, on-the-fly decisions.

1. Introduction

As projects grew in size, they also grew in complexity making effective project planning a hard task. It is mainly due to several interrelated factors that all have to be taken into consideration to predict in an accurate and precise way the cost and the duration of the project [1].

Many attempts have been conducted to improve the project scheduling prediction [2-6], and each of them endeavors to offer an optimized schedule for a given project. Simulation is one of the techniques that has been successfully applied to project planning, e.g. in construction [7], where they use combined discrete-event/continuous simulation.

Duration of tasks in many project schedules cannot be modeled as deterministic [8], i.e. using fixed numbers. Consequently, probability distribution functions need to be utilized to describe durations of tasks. Even with this assumption, the obtained simulation models are still considered limited and non realistic. They fail to take into account the different interrelated factors and uncertainty that in practice lead to plan changes. As stated by Joslin and Poole, "*the simulation will be unrealistic if the plan is static"* [1].

Human allocation uncertainty is seen as an example of a series of factors that can lead to a plan change. During the project run, based on the specific situations a team could be assigned a task that was originally assigned to another team if the latter one is unavailable (and the former team is available). The distribution function used to reflect the duration of the task most probably will be different to match the properties of the new task executers. Usually, teams have different levels of expertise which will in turn affect durations of tasks they perform. A practical simulation model should handle this dynamic aspect of a plan and anticipate possible changes.

In this paper we move one step towards the more realistic modeling of project schedules. We allow both duration and sequence of tasks to be variable – based on available resources and depending on various on-the-fly decisions by participants on the project. We simulate project schedules both with and without the possible on-the-fly decision scenarios. The objective of this is to study the effects of such changes in the project schedule and show their significance. Furthermore, we aim to provide an approach for accurate prediction of project schedule duration given the afore-described circumstances.

We use Gantt charts and state-transition diagrams for modeling project schedules. We extend both formalisms to model what we term as *state-dependent (floating) task*. Floating task represents a task that models uncertainty in human resources allocation and is a subject to various on-the-fly decisions. We chose the proxel-based method for simulating the project schedules, as it has been successfully applied to this area [9, 10] and can provide highly accurate complete results. These attempts did not address the uncertainty issues of on-the-fly decisions and resource allocation.

Input probability distribution functions can be fitted based on historical data for similar tasks and situations and may be adapted to concrete situations of projects. The estimation process would, obviously, require a high level of expertise.

The remainder of this paper is structured as follows: In the next section we provide an overview of problems faced in the project scheduling simulation and we describe the proxel-based method. Further, we introduce the floating task model and our running example as a software development process. Then, we illustrate our simulation scenarios as well as our simulation results followed by a discussion on the importance of uncertainty and on-the-fly decision modeling in simulation model. Finally, we conclude.

2. Problem Definition

A project consists of a number of tasks (activities) where a predefined set of tasks has to be processed in order to complete the project. The tasks are in fact related by two constraints:

1. Precedence constraints: usually in a project development tasks cannot be undertaken in any order and some tasks cannot start unless others have been already completed; and
2. Resource sharing: performing tasks requires efficient resources' management. Such resources may include financial resources, inventory, human skills, production resources, information technology (IT), etc.

The incorporation of *uncertainty* into project planning and scheduling has resulted in numerous research efforts, particularly focusing on uncertainty in task duration or cost [4]. In our case, we are interested in the effect of human uncertainty factor, both in terms of on-the-fly decisions and resource allocation, on the duration of a project.

As a running example, we consider a software development project. Typical requirements descriptions might include the task lists for people, and allocation schedules for resources.

Workforce allocation is seen as an important step in any software project management. It is the phase where all relevant elements of the software development process are taken into consideration for allocating software developers to the different project tasks [11].

While an initial allocation of software developers is calculated based on initial requirements, it is frequent that workforce adjustments during project performance becomes necessary for several reasons : (1) projections recalculation, based on the current workforce size, and the current development productivity [11], (2) number of remaining requirements to be implemented, and (3) requirements volatility.

Because of the above-mentioned reasons, it may happen that a team is assigned to a task that was originally assigned to another team during the workforce adjustment. Such scenario is in fact not easy to consider during the project scheduling. First, it is difficult to know when such adjustment will happen. This decision will be taken on-the-fly. Second, and more importantly, different teams have distinct expertise. Put in other words, the time that would take team A and team B to finish a task is not necessarily the same. It is frequent that the one of the teams may need to acquire the necessary expertise to achieve the task. When predicting the duration of project schedules, such scenarios should be considered.

The objective of our approach is to compute the probability distribution function of the duration of the project schedule taking into consideration human resource allocation uncertainty and typical on-the-fly decision behaviors. Our target is to observe the duration of the project in conjunction with on-the-fly decision scenarios for maximizing utilization of available personnel resources to achieve business goals.

3. The Proxel-Based Method

The proxel-based method [12, 13] is a relatively novel simulation method, whose underlying stochastic process is a discrete-time Markov chain [14] and implements the method of supplementary variables [15]. The method, however, is not limited to Markovian models. On the opposite, it allows for a general class of stochastic models to be analyzed regardless of the involved probability distribution functions. In other words, the proxel-based method combines the accuracy of numerical methods with the modeling power of discrete-event simulation.

The proxel-based method is based on expanding the definition of a state by including additional parameters which trace the relevant quantities in one model through a previously chosen time step. Typically this includes, but is not limited to, age intensities of the relevant transitions. The expansion implies that all parameters pertinent for calculating probabilities for future development of a model are identified and included in the state definition of the model.

Proxels (stands for probability elements), as basic computational units of the algorithm, follow dynamically all possible expansions of one model. The state-space of the model is built on-the-fly, as illustrated in Figure 1, by observing every possible transiting state and assigning a probability value to it (*Pr* in the figure stands for the probability value of the proxel). Basically, the state space is built by observing all possible options of what can happen at the next time step. The first option is for the model to transit to another discrete state in the next time step, according to the associated transitions. The second option is that the model stays in the same discrete state, which results in a new proxel too. Zero-probability states are not stored and, as a result, no further investigated. This implies that only the truly reachable (i.e. tangible) states of the model are stored and consequently expanded. At the end of a proxel-based simulation run, a transient solution is obtained which outlines the probability of every state at every point in time, as discretized through the chosen size of the time step. It is important to notice that one source of error of the proxel-based method comes from the assumption that the model makes at most one state change within one time step. This error is elaborated in [13].

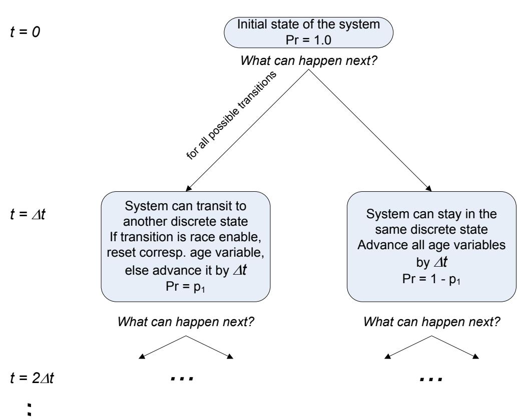

Figure 1. Illustration of the development of the proxel-based simulation algorithm

Each proxel carries the probability of the state that it describes. Probabilities are calculated using the instantaneous rate function (IRF), also known as hazard rate function. The IRF approximates the probability that an event will happen within a predetermined elementary time step, given that it has been pending for a certain amount of time τ (indicated as 'age intensity'). It is calculated from the probability density function (f) and the cumulative distribution function (F) using the following formula:

$$\mu(\tau) = \frac{f(\tau)}{1 - F(\tau)} \qquad (1)$$

As all state-space based methods, this method also suffers from the state-space explosion problem [16], but it can be predicted and controlled by calculating the lifetimes of discrete states in the model. In addition, its efficiency and accuracy can be further improved by employing discrete phases and extrapolation of solutions [17]. More on the proxel-based method can be found in [13].

4. State-dependent (floating) task

4.1 Vital vs. non-vital tasks

To formalize uncertainty we define the highly uncertain *state-dependent task*, for which we allow any relevant parameters determine its duration. We term this type of task as *floating* task. Its duration probability distribution is a complex function that among other factors depends also on the team that performs the task (its previous training, number of participants, etc.).

To introduce human decision uncertainty factors in project scheduling, we classify tasks into two categories, i.e. vital and non-vital, depending on their importance for the success of the project and the risk strategy of the project, i.e.:

1) Vital tasks: these tasks are estimated as critical for the success of the project. They are assigned only to *experienced professionals* to reduce the risk of their failure. Consequently, vital tasks are assigned a single team responsible for their implementation (fixed resource allocation strategy).
2) Non-vital tasks: these tasks are estimated as secondary for the success of the project. Non-vital tasks can be assigned to more than one team. Any of the teams that become available can implement it in order to optimize the project duration and maximize resource utilization. Under certain circumstances, non-vital tasks can be cancelled as well. In general, non-vital tasks invite various on-the-fly decision scenarios.

Whether a task is vital or non-vital tasks can be determined from the project requirements.

If we consider our running example, it is well known in software requirements management that users and stakeholders establish priorities to the feature set. Typical priority levels are: *critical*, *important*, and *useful* [18]. When simulating project schedules, we propose to categorize the set of prioritized features as vital and non-vital tasks. It is obvious that a critical feature with high risk cannot be seen as non-vital task since non-vital task may be even cancelled while a useful feature can be seen as a non-vital task. Introducing feature risk level as factor to decide about the categorization of the different features into vital and not vital tasks is out of the scope of this paper. Such issues are seen as part of our future work.

4.2 Simulation Model: Floating task

In the following we present our sample model that we use to demonstrate our approach. The Gantt chart of the sample software development project schedule is shown in Figure 2. Each of the tasks corresponds to a software requirement.

The project schedule has two software developer teams assigned (A and B) and commences by running two tasks (T1 and T2, both vital) in parallel. There is a third task (T3, non-vital) that originally needs to be executed by team A after they complete task T1, as team B is not trained to respond to this type of task. Only once all three tasks are finished the project can proceed to task T4 (vital), which is executed by both teams. However, it is a very typical scenario that if task T2 is completed in a very short time and T1 is *far from finished* then team B starts working on task T3 (since it is a non-vital task) instead of staying idle and task T3 waiting for team A.

Furthermore, if team A completes T1 *shortly after* team B has started to work on T3, then T3 can be cancelled (because T4 is vital and thus more important to be completed by both teams). Else, team A has to stay idle until team B completes task T3. This implies that the transition associated with the completion of task T1 by team A while team B is processing T3, is conditioned on the history (more specifically the age intensities) of the model.

In our sample model we observe two possible decision scenarios, as described in the following:

a) If duration of task T2 performed by team B is “very short” then start T3 by team B.
b) If duration of T1 is “too long” and it completes “shortly after” team B started to work on T3, then T3 is cancelled and both teams start working on T4.

Apparently, there is fuzziness in the project schedule description (i.e. “far from finished”, “shortly after”) that requires adequate modeling. For that we use the following fuzzy function (however there is no limit as to what function can be used) [19]:

$$f(x) = \begin{cases} 0, x < a \\ \dfrac{x-a}{b-a}, a \le x \le b. \\ 1, otherwise \end{cases} \tag{2}$$

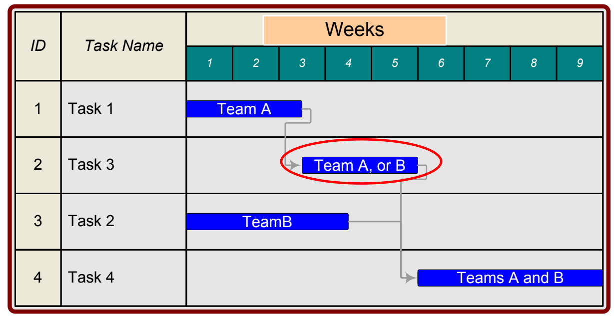

Figure 2. Gantt chart of the example model, floating task is encircled in red color

5. Proxel-based simulation of extended project schedules

In the following we provide the details of the proxel-based simulation of the sample project schedule that involves a floating task. This should serve as description of our approach through an example.

Each task in the model has a name, a priority level (vital or non-vital), a duration probability distributions with respect to possible team association, and a set of pre-requisite tasks. The proxel format of the state of the project schedule encompasses the following parameters:

- task vector $\{T_i\}$, where T_i is the task that team i is working on, or I for idle,
- age intensity vector $\{\tau_i\}$, for tracking the duration of tasks,
- probability value.

Thus the format of the proxel is as follows:

Proxel = (*Task Vector, Age Intensity Vector, Probability*)

The initial proxel, i.e. the proxel that marks the initial state of the system would be ((T1, T2), (0, 0), 1.0). It describes the situation in which team A is working on task T1, and team B on task T2 with a probability of 1.0. In the next time step the model can do each of the following developments:

a) Task T1 is completed,
b) Task T2 is completed, or
c) None of the tasks are completed

Resulting into the following three proxels:

a) ((T3, T2), (0, Δt), p_1)
b) ((T1, T3), (Δt, 0), p_2)
c) ((T1, T2), (Δt, Δt), 1 - p_1 - p_2)

In case (a), team A starts working on task T3, and also the corresponding age intensity is now reset to track the duration of T3. In case (b) team B takes over task T3, instead of sitting idle and waiting on team A to finish task T3. Case (c) shows the situation of both teams continuing what they have been doing before.

Because of the on-the-fly decision scenarios, both (T3, T2) and (T1, T3) can transit to (T4, T4). If T1 is completed shortly after team B has started working on T3, then the model transits to (T4, T4) with the completion of T1. Else, it waits for team B to complete T3 before transiting to (T4, T4). For generating each new proxel, the durations of tasks in progress need to be investigated for the decision modeling.

The state-transition diagram of the sample project schedule is shown in Figure 3. As depicted with the extra wide arrow ⬀, when team A is working on T1 and team B on T3, the transition associated with the completion of T1 depends on the time that team B has already spent on working on task T3. If it was "too long" then team A will stay idle and wait for its completion. One the other hand, if team B has just started working on task T3, then it is interrupted and both teams start working on task T4 which leads to completing the project.

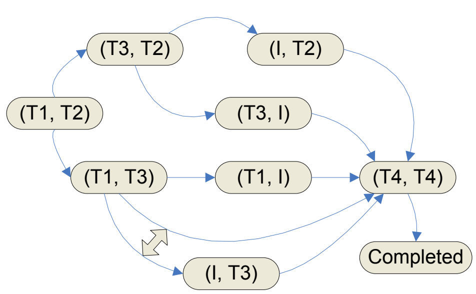

Figure 3. State-transition diagram of the project schedule

The algorithm that we have developed represents an extension of the original proxel-based method [12, 13]. In particular, the differences can be summarized as: *Before processing each transition, check all possibilities for possible flow changes based on proxel parameters. Generate subsequent proxels correspondingly.*

6. Experiments and Results

6.1. Experimental Environment

The experiments were run on a standard workstation with an Intel Core2Duo Processor at 2.0 GHz and 1 GB RAM. The choice for Δt was 0.1 and the simulation was run up to time $t = 20$. This implies that the number of simulation steps was 200.

The computation time for this experiment was ca. 4 seconds. In the following we present the results, i.e. the statistics that were calculated during this simulation experiment. The input data is provided in Table 1.

Table 1. Input data for the sample project schedule (N-Normal, U-Uniform)

Team/task	T1	T2	T3	T4
A	N(5, 1)		U(2, 4)	U(0.5, 2)
B		U(2, 10)	N(7, 1.2)	

6.2. Experiments

The goal of the experiments is to show the importance of modeling the effects of on-the-fly human decision behaviors on project schedules. For that purpose we first simulated the project schedule in an ideal scenario, i.e. excluding any intrusions during project running. Next, we simulated the project duration exposed to the hypothetical scenarios (both (a) and (b)) for on-the-fly project flow decisions. To study the effect of neglecting them, we compare both solutions and present the results with a chart.

In Figure 4, we can see the probability distribution of project duration for all combination of on-the-fly decision scenarios. In order to represent closely the effect of their modeling and simulation, Figure 5 shows the difference of the two probability distributions, with and without on-the-fly decision scenarios. The results show that in our sample model, it goes up to ca. 0.28, which is far from negligible.

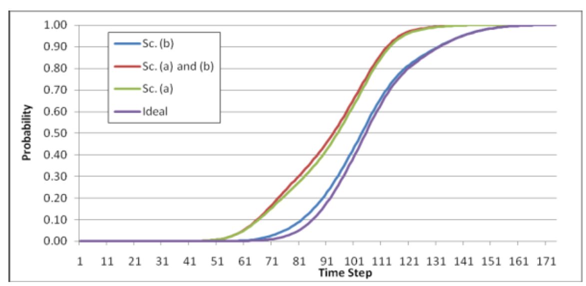

Figure 4. Probability distribution of the duration of the project schedule with the 4 possible combinations of scenarios

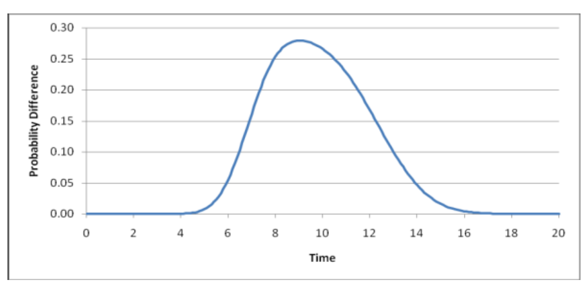

Figure 5. Project schedule duration probability distributions difference for the ideal and project schedule with both scenarios

7. Discussion

The approach that we presented allows a higher degree of uncertainty in project schedules to be modeled and simulated. The uncertainty that we observe is in terms of duration of tasks, task allocation, as well as arbitrary on-the-fly decisions that influence the workflow. We all witness that these things happen almost every time and in every project. Thus, simulation models need to consider them in order to obtain accurate measures for the duration of project schedules. Very often, these factors are neglected, and by our example we showed what difference they can make. In our example model, the probability difference for the completion of the project reached ca. 0.27, and this is still just a toy model. In real project schedules it can be more extreme and thus it has to be taken into account. The question that arises is how to obtain the numbers that represent and model these behaviors. We believe that they can be modeled by historical data and tracking from previous projects of similar types. In addition, expert knowledge and common sense can help to a great extent.

8. Summary and Outlook

This paper presents a more realistic project schedule simulation and modeling approach that allows for a high level of uncertainty. The purpose of this simulation model is: (a) to model the uncertainty of human resources allocation to the different project tasks and (b) to take advantage of this uncertainty to simulate various on-the-fly human decisions and their impact on the project duration.

To extend our work we plan to address the effect of these uncertainty factors on the productivity and budget, by adding value, effort and cost parameters. In addition, we intend to extend our simulation model to handle the effects of requirements volatility in software engineering.

References

[1] D. Joslin and W. Poole, "Agent-based simulation for software project planning," presented at Proceedings of the 37th conference on Winter simulation Orlando, Florida 2005.

[2] Z. Jing-wen and S. Hui-fang, "Multi-Mode Double Resource-Constrained Time/Cost Trade-Offs Project Scheduling Problems," presented at International Conference on Management and Service Science, 2009. MASS '09. , 2009.

[3] W. Huang, L. Ding, B. Wen, and B. Cao, "Project Scheduling Problem for Software Development with Random Fuzzy Activity Duration Times," 2009.

[4] W. Herroelen and R. Leus, "Project scheduling under uncertainty: Survey and research potentials," *European Journal of Operational Research*, vol. 165, pp. 289-306, 2005.

[5] J. A. Arauzo, J. M. Galán, J. Pajares, and A. López-Paredes, "Multi-agent technology for scheduling and control projects in multi-project environments. An Auction based approach," *Inteligencia Artificial*, vol. 42, pp. 12-20, 2009.

[6] M. J. Sobel, J. G. Szmerekovsky, and V. Tilson, "Scheduling projects with stochastic activity duration to maximize expected net present value," *European Journal of Operational Research*, vol. 198, pp. 697-705, 2009.

[7] S. M. AbouRizk and R. J. Wales, "Combined discrete-event/continuous simulation for project planning," *Journal of Construction Engineering and Management*, vol. 123, pp. 11-20, 1997.

[8] O. Perminova, M. Gustafsson, and K. Wikström, "Defining uncertainty in projects–a new perspective," *International Journal of Project Management*, vol. 26, pp. 73-79, 2008.

[9] C. Isensee and G. Horton, "Proxel-Based Simulation of Project Schedules," 2004.

[10] B. Rauch-Gebbensleben, K. Dammasch, and G. Horton, "Generierung und Visualisierung des Zielkorridors in Projektplänen mit stochastischen Einflussgrößen," in *Simulation und Visualisierung 2008*. Magdeburg, 2008.

[11] D. Pfahl and K. Lebsanft, "Using simulation to analyse the impact of software requirement volatility on project performance," *Information and Software Technology*, vol. 42, pp. 1001-1008, 2000.

[12] G. Horton, "A new paradigm for the numerical simulation of stochastic Petri nets with general firing times," *Proceedings of the European Simulation Symposium*, 2002.

[13] S. Lazarova-Molnar, "The Proxel-Based Method: Formalisation, Analysis and Applications," in *Faculty of Informatics*, vol. Ph.D. Magdeburg: University of Magdeburg, 2005.

[14] W. J. Stewart, *Introduction to the Numerical Solution of Markov Chains.*: Princeton University Press, 1994.

[15] D. R. Cox, "The analysis of non-Markovian stochastic processes by the inclusion of supplementary variables," *Proceedings of the Cambridge Philosophical Society*, vol. 51, pp. 433-441, 1955.

[16] F. J. Lin, P. M. Chu, and M. T. Liu, "Protocol verification using reachability analysis: the state space explosion problem and relief strategies," *ACM SIGCOMM Computer Communication Review*, vol. 17, pp. 126-135, 1987.

[17] C. Isensee and G. Horton, "Approximation of Discrete Phase-Type Distributions," *Proceedings of the 38th annual Symposium on Simulation*, pp. 99-106, 2005.

[18] D. Leffingwell and D. Widrig, *Managing Software Requirements: a use case approach*: Pearson Education, 2003.

[19] M. Demirci, "Fuzzy functions and their fundamental properties," *Fuzzy Sets and Systems*, vol. 106, pp. 239-246, 1999.