A FRAMEWORK FOR ENHANCED PROJECT SCHEDULE DESIGN TO AID PROJECT MANAGER'S DECISION MAKING PROCESSES

Sanja Lazarova-Molnar[(a)], Rabeb Mizouni[(b)], Nader Kesserwan[(a)]

[(a)] Faculty of Information Technology, United Arab Emirates University, United Arab Emirates
[(b)] Department of Software Engineering, Khalifa University, United Arab Emirates

[(a)]{sanja,nkesserwan}@uaeu.ac.ae, [(b)]rabeb.mizouni@kustar.ac.ae

ABSTRACT

Good schedule increases the chances of a project meeting its goals. The most popular formalisms for describing project schedules are very rigid and inflexible in modeling changes due to uncertainties. In this paper we describe a framework to support enhanced project schedule design. The proposed framework is based on the Enhanced Project Schedule (EPS) model. In addition to an initial Gantt Chart, EPS allows definition of Remedial Action Scenarios (RAS), which contain guidelines of actions to consider when uncertainties arise. This creates a dynamic and evolving schedule. It is meant to guide the project manager in the decision making processes throughout the project implementation. The process of selection of the remedial action scenario is an optimization one, based on simulation. We illustrate the dynamics of the EPS design framework by an example.

Keywords: project schedule, proxel-based simulation, remedial action scenarios, uncertainty

1. INTRODUCTION

During the past few decades project management has evolved into a discipline that studies planning, scheduling and controlling of activities that directly contribute to the achievement of project's objectives.

The pressure of time-to-market along with the increasing complexity of present-day projects, have contributed project management to become one of the main factors for projects success. A growing number of companies use various advanced project management tools and methods to ensure the project quality expected by customers, delivered within reasonable deadlines and at the lowest possible cost.

Many attempts have been conducted to improve the project scheduling prediction (Herroelen and Leus 2005; Arauzo, Galán et al. 2009; Huang, Ding et al. 2009; Jing-wen and Hui-fang 2009; Sobel, Szmerekovsky et al. 2009). Many of them are based on analytical models and simulation. Tools, such as Microsoft Project and Primavera Project Planner, are typically suggested to help managers in planning and controlling their projects. Existing frameworks and methods, however, fail, or, are insufficient; to answer the real needs of a project. The models developed still suffer from many limitations that often make them not representative to real world situations. Typically, project schedules are described in very strict terms, using Gantt charts or PERT.

In real life, even small projects face risks and may, consequently, deviate from their original plans. As a consequence, even good projects can fail (Matta and Ashkenas 2003). For instance, in the software industry it has been reported (Denning and Riehle 2009) that approximately one-third of software projects fail to deliver anything, and another third deliver something workable but not satisfactory. In order to have a more realistic and effective project scheduling, management frameworks need to incorporate uncertainties on the one hand, and guide the managers to what actions to take when such uncertainties arise. This is the issue that we address in our paper, i.e. to lay out a strategy to create an *optimal enhanced project schedule*. Our objective is to answer the needs of managers by providing a framework that helps the generation of *a more realistic and insightful* project planning.

The proposed framework supports flexible and efficient project schedule modeling and simulation. It combines: (a) a novel model for describing project schedules in a more realistic way, accommodating uncertainties, and (b) facilities for model's simulation and assessment with respect to predetermined project goals. The objective of the framework is to provide managers with answers to the following types of questions:

1) What is the best Remedial Action Scenario (RAS) to adopt if some uncertainties arise during the implementation of the project?

2) What are the features that can be implemented within the deadline of the projects?

3) What are the best and robust deadlines to consider that take into consideration the deviation from the original scheduling because of uncertainties.

As shown in Figure 1, our framework is based on two main modules, and two supporting ones:

1) Multi-RAS EPS Proxel-Based Simulator is a simulator based on the proxel-based simulation method.

2) Result Visualization Module: responsible for visualizing and interpreting the results of the simulation with respect to the goals specified by the manager.

3) User Interface Module that supports and facilitates the input of project schedules.

4) A Data Storage Module that manages project schedule data.

The rest of the paper is organized as follows. In the next section we describe the EPS model. In Section 3 we present the framework that supports the generation of EPS models and we describe the different modules, focusing on the key ones. Section 4 demonstrates the idea of the framework by an example. Finally, Section 5 concludes the paper.

2. WHAT IS AN ENHANCED PROJECT SCHEDULE?

To illustrate the concept of an enhanced project schedule, we provide an example that displays side-by-side a standard Gantt chart and an EPS, as shown in Figure 2. For comparison, Figure 2(a) illustrates a simple project schedule, modeled using a classical Gantt chart. The project schedule consists of four tasks (Task1, Taks2, Task3, and Task4) and two available teams (Team A and Team B). All tasks have predefined executors leading to one possible scenario of execution. Such model is in fact rigid and it is not able to anticipate the occurrence of any unpredictable events.

Figure 2(b) illustrates the EPS. While having the same number of tasks and teams, two majors features are added:

1) “floating task” (Task 2), which is a non-vital task that can be executed by any of the two teams, albeit with different duration distribution functions (based on teams’ expertise).

2) fuzzily described guidelines, provided below the schedule, which are meant to accompany the project schedule as RAS (remedial action scenario).

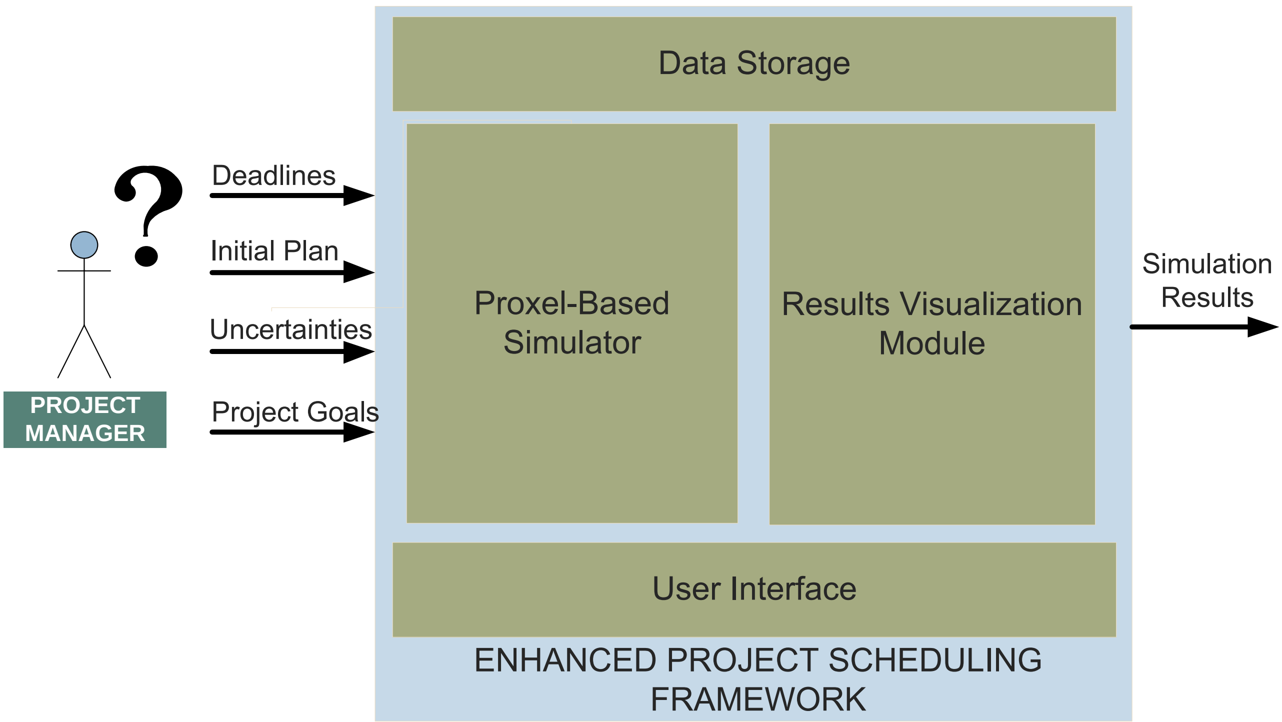
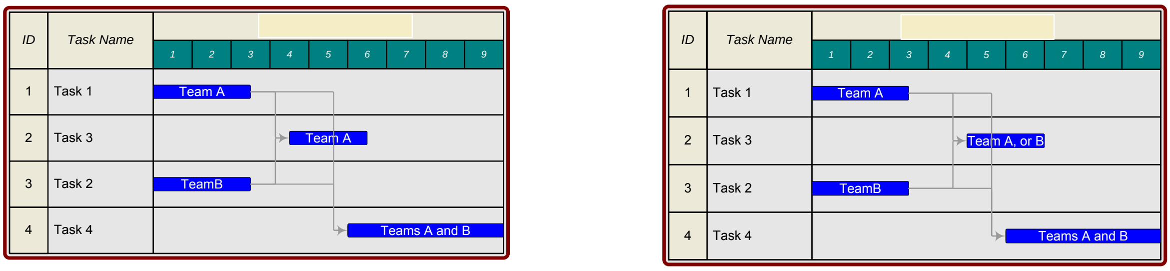


Figure 1: EPS Design Framework Architecture

(a) No additional rules

(b) EPS with a Remedial Action Scenario



1. If duration of Task 2 performed by team B is "very short" then start Task 3 by team B.
2. If duration of Task 1 is "too long" and it completes "shortly after" team B started to work on Task 3, then Task 3 is cancelled and both teams start working on Task 4.

Figure 2: Illustration of: a) Classical Gantt Chart and b) Enhanced Project Schedule with RAS

In our previous work (Lazarova-Molnar and Mizouni 2010; Lazarova-Molnar and Mizouni 2010) we successfully modeled and simulated the type of scenarios described in Figure 2(b). There, we also developed an approach to analyze and simulate the effects of the uncertainties and remedial actions on the duration of project. As expected, on-the-fly decisions make a significant difference in the duration of the project and need to be considered and, if possible, pre-determined. To account for resource re-allocations we have also defined a new type of tasks, which we termed as "floating task". This task was a typically a non-crucial task for the success of the project, which could be implemented by a number of teams, albeit with different duration distribution functions, and based on their availabilities.

2.1. EPS Model Description

We propose the definition of a schedule to include the uncertainties that can arise and their quantification using statistical probability distributions. In addition to this, we formalize the remedial actions that managers can take. Every schedule along with the set of remedial actions (RAS) creates what we term as: *enhanced project schedule* (as shown in Figure 3). The RAS consists of a set of fuzzy *if-then* production rules. These rules make the project evolving and thus, the sequencing of tasks, dynamic and changing. Once the enhanced project schedule is designed, we simulate each RAS using the proxel-based method and pick the best one based on the success criteria for the project. "The probability that the project is delivered before deadline" and "the probability that the project is implemented within this budget" are examples of success criteria.

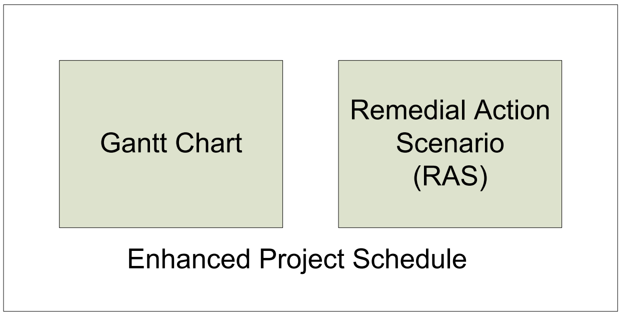


Figure 3: Enhanced Project Schedule Components

Let us take the example of a simple software development project schedule, subject to various uncertainties. The enhanced schedule would consist of a Gantt chart, where each task that corresponds to a requirement implementation, is associated with a probability distribution function for its duration, as well as a set of fuzzy rules that describe the remedial strategy under certain conditions (e. g. if task A finishes in a very short time than proceed to task B, else skip task B). The set of fuzzily specified guidelines are obtained by simulating a set of possible RAS, and accordingly selecting the most optimal one (similar to the simple example presented in Figure 2(b)).

Once the simulation of the chosen RAS provides good results with respect to project goals (e.g. complete as many tasks as possible, complete in as short time as possible, or minimize budget), the resulting EPS is communicated to the project manager to aid his/her decision making process. We see the fuzziness as a great advantage as it leaves a certain degree of freedom to the project manager as well, to involve his/her knowledge/perceptions he/she might have.

The proxel-based simulation method allows for a great flexibility in schedule description and provides solutions taking into account anticipated uncertainties. This helps us in picking the best RAS that specifies the optimal set of recommended remedial actions when uncertainties occur.

2.2. EPS Formal Model

An Enhanced Project Schedule (EPS) is described as follows:

$$EPS = (A, P, T, D, W, F, IGC)$$

- $A = \{A_1, A_2, \dots, A_n\}$, set of tasks, where each task corresponds to a task in the project schedule,
- $P = \{P_1, P_2, \dots, P_m\}$, set of precedence constraints, that are actually tuples of two tasks where the completion of the first one is a pre-requirement for commencing the second one, e.g. (A_x, A_y) would mean that completing of A_x is a pre-requirement for beginning A_y,
- $T = \{T_1, T_2, \dots, T_l\}$, set of teams available for the execution of the project,
- $D = \{d_1, d_2, \dots, d_s\}$, set of probability distribution functions that correspond to duration of tasks performed by the competent teams,
- $W = \{w_1, w_2, \dots, w_t\}$, set of mappings of distribution functions to competent teams and tasks,
- $F = \{f_1, f_2, \dots, f_r\}$, set of fuzzy rules that define the remedial action scenario,
- $IGC - Initial\ Gantt\ Chart$, initial sequencing of tasks that satisfies the set of precedence constraints provided by P,

where $P \subseteq A \times A$, and $W \subseteq A \times M \times D$. Also, $A = A^c \cup A^n$, where A^c is the set of cancelable tasks and A^n is the set of non-cancelable tasks. Cancelable task is a task that is non-vital for the success of the project, and thus, not compulsory, however, useful for the value of the project. Non-cancelable tasks are the ones that are crucial for the success of the project. This differentiation is important for the realistic simulation of project schedules.

Each fuzzy rule is made up of two parts: *condition* and *action*, formally expressed as "$condition \Longrightarrow action$". Conditions can be described either by using strict terms, or fuzzy ones. An action can typically be canceling or interrupting some of the tasks, or one of the various types of rescheduling. This is the fact that makes our schedule description evolving, rather than rigid and inflexible. Two examples of fuzzy rules are:

$$A_x\ takes\ too\ long \Rightarrow cancel\ A_y$$

or

$$A_x\ completes\ quickly\ after\ A_y \Rightarrow cancel\ A_z.$$

Both are examples for typical proceedings during project execution. However, in our approach we formalize their modeling, assessment and quantitative evaluation. This makes it straightforward to compare various RAS, as well as test for the best RAS to counteract uncertainties, as described by F. Note that F can be an empty set too, which would imply sticking to the original project schedule provided by *IGC*. Once an optimal remedial action scenario is selected, it is associated with the initial project schedule and handed to the project manager as a decision making aid. The goal of the proposed framework is to support the generation of the EPS. This process is further demonstrated by a simple example in Section 4.

2.3. Multi-RAS EPS

To facilitate the generation of the optimal EPS, the framework needs to analyze a number of various RAS that could potentially accompany a given EPS. For this purpose, we define the term Multi-RAS EPS (MEPS) as follows:

$$MEPS = (A, P, T, D, W, F', IGC),$$

where the only difference to the standard EPS is that $F' = \{F_1, F_2, \dots, F_q\}$ represents a set of RAS, where each $F_i, i = 1..q$, is a defined as a set of fuzzy rules. This defines the central input to the framework.

3. THE EPS DESIGN FRAMEWORK

Main functionalities of the proposed framework for EPS design are the following:

1. Support the expressive description of EPS,
2. Support definitions of project goals (e.g. minimize duration; maximize number of completed tasks; etc.),
3. Run simulations for a single project schedule in combination with a number of RASs (Multi-RAS EPS), and
4. Select the RAS that best meets the specified project goals to accompany the initial project schedule to yield the final EPS.

The simulation method of choice is the proxel-based simulation (Horton 2002; Isensee, Lazarova-Molnar et al. 2005) as it is highly flexible and provides high accuracy. Its additional advantage is that the simulation is carried out directly, based on the user model, i.e. EPS, without building the state space prior to that. The resulting, simulation-based calculated, optimal EPS will definitely take into account many of the uncertainty factors, thus reducing the risk in the project. In addition, it will provide managers with more insight and guidance when making decisions during project's implementation. A high-level diagram of the data-flow process that underlies the EPS supporting framework is presented in Figure 4. It shows that the inputs to the program are the Multi-RAS EPS and the Project Goals, and it produces an Optimal EPS as a final product.

In the following, we provide detailed description of the most complex module of the framework, i.e. the

EPS Proxel-Based Simulator as previously mentioned in Section 2, and a brief description of the remaining modules.

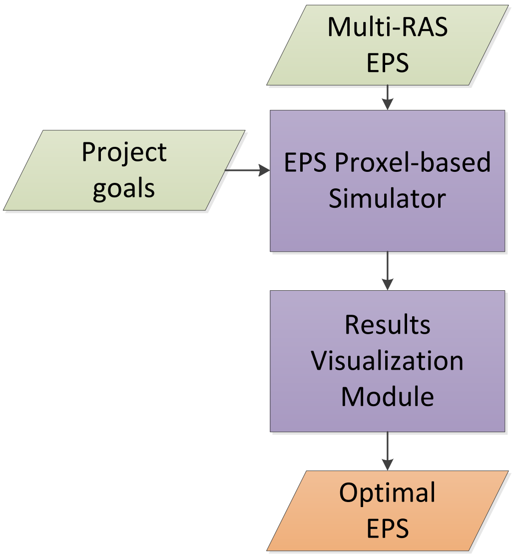


Figure 4: High-level diagram of the framework

3.1. EPS Proxel-Based Simulator Module

The Proxel-Based Simulator is the key-module of the framework. This is the module that performs simulation of the provided Multi-RAS EPS. During simulation, statistics that correspond to project's goals are collected. As previously stated, our simulation method of choice is the proxel-based simulation (Lazarova-Molnar 2005).

The proxel-based method is a simulation method based on the method of supplementary variables (Cox 1955). It was introduced and formalized in (Horton 2002; Lazarova-Molnar 2005). The advantages of the proxel-based method are its flexibility to analyze stochastic models that can have complex dependencies and the accuracy of results, which is comparable to the accuracy of numerical solvers (Stewart 1994).

The proxel-based method expands the definition of a state by including additional parameters which trace the relevant quantities in one model following 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.

In order to apply the proxel-based simulation algorithm, this module needs to process the information contained in the input file, i.e. the Multi-RAS EPS. In summary, it contains the following information:

- Maximum simulation time,
- Time step,
- Task information,
- Team information,
- Distribution functions in use,
- Mappings of distribution functions to teams and tasks,
- Multiple RAS,
- Deadline, and
- Initial state.

The proxel-based simulation of a given project schedule in combination with each provided RAS is the core element of the tool. Algorithm 1 provides more details of how this is performed. It describes the dynamics of the proxel-based simulation for a single-RAS EPS. This is further repeated for each provided RAS. The basic computational unit, i.e. the proxel, for each EPS is formed based on the information in the input file. The general simplified proxel format is the following:

$$Proxel = (State, t, Pr)$$

where:

$$State = (Task\ Vector, Age\ Vector, Completed\ Tasks)$$, and

- *Task Vector* is a vector whose size is equal to the number of teams available and records the task that each team is working on,
- *Age Vector* tracks the length that each team has been working on the task specified in the *Task Vector*, correspondingly,
- *Completed Tasks* stores the set of completed tasks,
- *t* is the time at which the afore-described state is observed, and
- *Pr* stores the probability that the schedule is in the afore-specified state at time *t*.

Algorithm 1 demonstrates the on-the-fly building of the state-space of the project schedule model. Thus, there is no need for any pre-processing to generate the state-space. It is directly derived from the input file specification. The initial state proxel is derived from the initial state that is specified in the input file as well.

The algorithm operates by using two interchangeable data structures, Proxel_Tree[0] and Proxel_Tree[1], that store the proxels from two subsequent time steps (regulated by the *switch* variable). If two proxels represent the same state, there is only one proxel stored, and their corresponding probabilities are summed up.

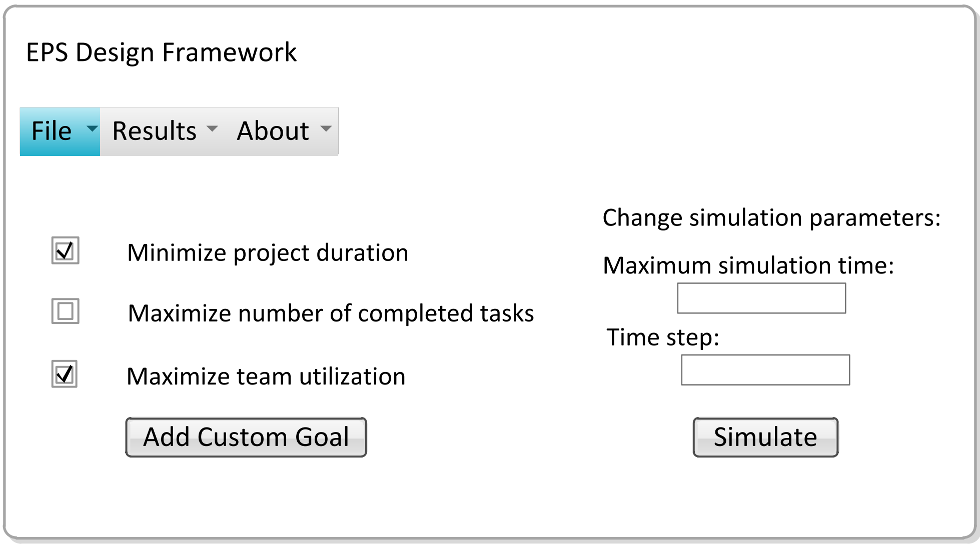


Figure 5: EPS Design Framework GUI prototype

Algorithm 1: Proxel-based simulation of enhanced project schedules

```
Input: EPS, Project Goals
Output: Simulation Results
switch = 0
insert Initial State Proxel in the Proxel_Tree[switch]
switch = 1 - switch
while (maximum simulation time has not been reached)
{
  px = get_proxel(Proxel_Tree[switch]);
  for (each task in the Task Vector(px))
  {
    check task precedence & team availability;
    generate next state S;
    compute probability for S in computed_prob
    search for the S in the Proxel_Tree[1-switch];
    if (S found)
    {
        px1 = found_proxel(S);
        probability(px1) = (probability(px1) ) +
            computed_prob;
    }
    else
    {
        generate new proxel px2(S);
        insert proxel in Proxel_Tree[1-switch];
    }
  delete px from Proxel_Tree[switch];
  increase simulation time by one time step;
  calculate statistics with respect to project goals;
  switch = 1- switch;
  }
}
```

3.2. Supporting Modules

The remaining modules, i.e. the Results Visualization Module and the two supporting ones, are trivial. The Visualization Module is charting the (transient or steady-state) solutions of the simulation with respect to project goals. An example of such solution is provided in the following Section 4.

The Graphical User Interface Module facilitates file-based and graphical input of Multi-RAS EPS, along with the set of project goals. The project goals are meant to be selected from a list of most commonly used ones. The list would include project goals as:

- minimize duration,
- maximize number of tasks completed, etc.

as well as allow the user to specify custom goal by using a scripting language. To illustrate our idea, a prototype of the framework GUI is shown in Figure 5.

The Data Storage Module ensures efficient memory manipulation and stores the statistics and intermediate solutions of the simulation experiments.

4. EXPERIMENTS

To demonstrate the proposed framework, we demonstrate the processing of an example Multi-RAS EPS. The example EPS contains 4 tasks, identified as: Task 1, Task 2, Task 3, and Task 4. Each task can be performed by one of the two teams: Team A or Team B. Tasks 1, 2 and 3 have fixed human resource allocation, i.e. performing team, and Task 4 is a *cancelable floating task*, and can be performed by either team A or B. The initial Gantt chart (*IGC*) of the sample project schedule is shown in Figure 6, where the green-colored tasks are cancelable and the team capable of carrying out task is labeled on the task itself. In addition to this the project schedule has a predefined deadline Δ.

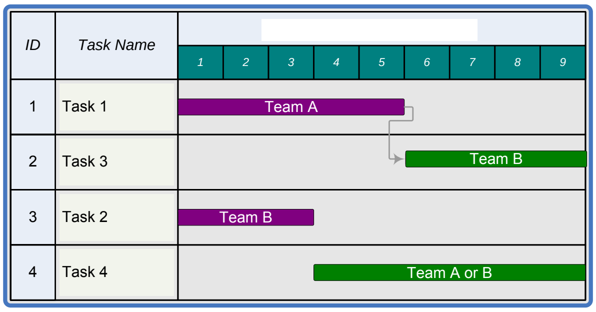


Figure 6: Example of an Initial Gantt chart of the example project schedule (the green-colored tasks are cancelable)

The multi-RAS enhanced project schedule features three RAS. In our case we choose among the two remedial action scenarios (*a* and *b*) and the default sequencing (seen as an empty set of fuzzy rules, *c*), which are defined as follows:

a) *If the duration of tasks 1 or 2 is close to the deadline Δ, then do not start working on any of the tasks 3 or 4 and do not interrupt the other team if they have already started to work on either of the latter two tasks.*

b) *If the team assigned to a certain non-floating and cancelable task (Task 3 in our case) is unavailable at the time it can be initiated, then cancel the task.*

c) *No guidelines are provided and the manager is instructed to follow the original schedule.*

The performance measure, according to which we assess the three RAS, is:

- *"the probability of completing the project before the deadline"*

The project goal that is supported by this performance measure is defined as "Complete the project before the deadline". The simulation targets to discover the RAS that yields the best performance, given the constraints of the initial project schedule. For this purpose, the EPS Proxel-Based Simulator Module runs the proxel-based simulation that collects the statistics that answer our question.

4.1. Input File Specifications

In the following we describe and explain the input file specification of the example model, which is shown in Figure 7.

```
#tasks
1 Task1 () 10 false false
2 Task2 () 05 false false
3 Task3 (1) 10 true false
4 Task4 () 10 true true
#teams
10 A
20 B
#distributions
100 U (2.0,10.0)
200 N (7.0,1.0)
300 U (2.0,8.0)
400 W (3.5,1.5)
500 U (2.0,5.0)
#ttd
1 10 100
2 20 200
3 20 300
4 10 400
4 20 500
#ras
(a)
1 fuzzy1 (11.25,15.0) C 3
1 fuzzy1 (11.25,15.0) C 4
2 fuzzy1 (11.25,15.0) C 3
2 fuzzy1 (11.25,15.0) C 4
(b)
3 non_avail (20) C 3
(c)
-
#deadline
20
#initial
(1,2)
```

Figure 7. Example Input File

The input file consists of:

1) Definition of all tasks and their duration probability distribution function,
2) Definition of all fuzzy functions in use,
3) Deadline of the project, and
4) Initial state(s)

The input file contains all parameters that are listed in Section 3.1. In the example case there are 4 tasks. Each task is specified by the following parameters: *Task ID*, *Task Name*, *Preceding Tasks*, *Cancelable*, and *Floating*, specified in the same order. Each team is specified by: *Team ID* and *Team Name*. Distributions are specified by: *Distribution ID*, *Distribution Type*, and *Parameters*. Each team-task-distribution mapping contains three values, i.e. the id's of the team, task and distribution that are connected to form the mapping. As specified in the input file, the values of the parameters of the duration distribution functions are:

- Duration of Task 1 ~ Uniform (2.0, 10.0)
- Duration of Task 2 ~ Normal (7.0, 1.0)
- Duration of Task 3 ~ Uniform (2.0, 8.0)
- Duration of Task 4, performed by:
  - Team A ~ Weibull (3.5, 1.5)
  - Team B ~ Uniform (2.0, 5.0)

Next, the definition of the RAS is provided. The fuzzy membership function that defines *fuzzy1* is defined in the framework as follows:

$$\mu(t,a,b)=\begin{cases} 0, & t<a \\ \dfrac{t-a}{b-a}, & a\le t\le b \\ 1, & t>b \end{cases}\cdot$$

As specified in the input file, the concrete fuzzy membership function is $\mu(t,11.25,15.0)$. The symbol C stands for “cancel” and the subsequent number specifies the task id of the task to be canceled. The *non_avail* function evaluates to true/false depending on the availability of team B (specified by its id, i.e. 20 as a parameter).

The framework allows custom specification of the fuzzy functions and actions. It is also extendable as to the type of actions that can be taken. Currently it features only “cancel”.

Finally the pre-determined deadline of 20 time units (as an important parameter) and the initial state of the EPS are provided. According to the latter one, the project begins by Team A working on Task 1 and Team B working on Task 2.

In addition to the EPS model specification, inputs to the framework are the simulation parameters (size of the time step and maximum simulation time) and project goals.

4.2. Proxel-Based Simulation Details

In the following we provide some insight in the proxel-based simulation of our example model to illustrate the simulation method.

The proxel-based simulation of the EPS commences with the initial state, as specified in the input file. This would create the following initial proxel:

(((1, 2), (0, 0), ()), 0, 1.0).

In the next time step, one of the three developments could be seen:

1) Team A completes working on Task 1,
   - implying that it can start working on Task 4, as the only possibility
2) Team B completes working on Task 2, and
   - implying that it can start working on Task 4, as the only possibility
3) Both teams continue working on the corresponding tasks.

This would create the following proxels, correspondingly:

1) (((4, 2), (0, Δt), (1)), 0, p1),
2) (((1, 4), (Δt, 0), (2)), 0, p2), and
3) (((1, 2), (Δt, Δt), ()), 0, 1-p1-p2).

Note that the age intensities for each task that is still being worked on in the next time step are updated accordingly.

4.3. Experimental Results

In the following we present the results of the simulation of our model. The proxel-based simulation provides complete results for any quantity of interest; in this case it is the probability function of the duration of the project (shown in Figure 8). The performance measures are not limited to this and it is provided for illustration only. In general, they can include any quantities that are relevant to project’s goals.

Simulation results provide us with an overview to aid the selection of the most suitable remedial action scenario with respect to project’s goals.

From the simulation results we can see that the best RAS that yields the highest probability of having the project completed before the deadline is RAS (a), closely followed by (b). Judging from this, we can conclude that the RAS (a) seems most favorable for this enhanced project schedule. Also, we can clearly see that the rigid RAS (c) which does not allow for any changes has the lowest probability of having the project completed before the deadline.

5. CONCLUSIONS

The framework presented in this paper demonstrates a vision and its implementation strategy of how to create a better project schedule, one that will provide more information and decision making guidance to project managers. This is what we term as EPS Design Framework. The output of the framework, i.e. the resulting EPSs is planned to be further used to support project managers in the decision making processes throughout project implementation. Decisions made in this way, based on the RAS recommendations that accompany the optimal EPS will not be solely based on human judgment, as it is the case in classical approach, but also based on sound models and their analysis.

In this paper we present the details of the framework and, in particular, the details of its core module, i.e. the Proxel-Based Simulator, for which we present the modified proxel-based simulation algorithm.

We believe that this is an effective way of designing schedules and it enhances the classical project schedule by allowing all available information to be utilized. Instead of having a static schedule, the remedial action scenarios make the schedule dynamic and evolving. In addition, our framework provides support for managers to incorporate their knowledge within the project schedules by simulating various possible RAS when new uncertainties occur. The

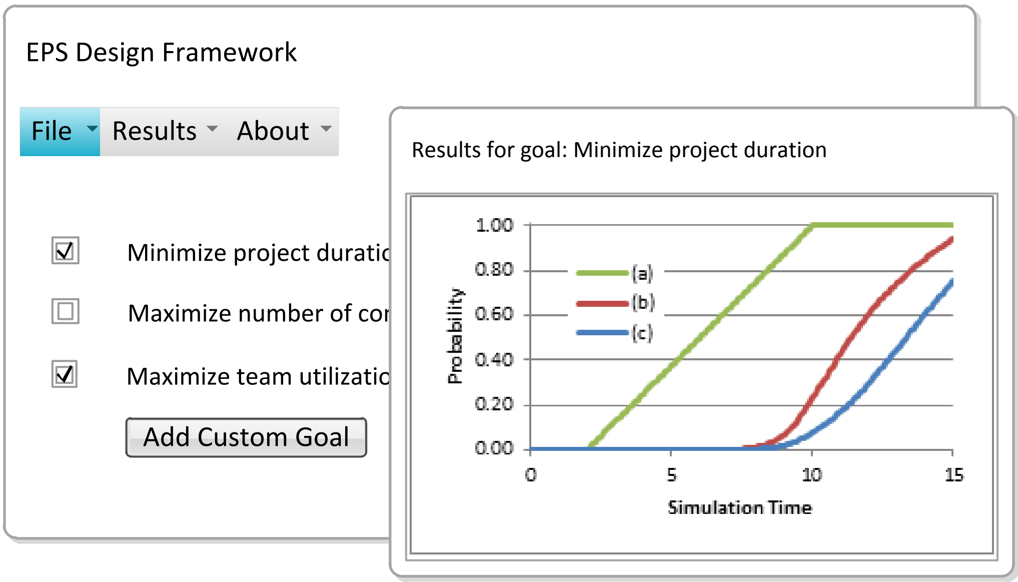


Figure 8: Probability of having the project completed for the three possible RAS

framework encourages analysis and deep thinking about project plans and, hence, supports the creation of a higher quality initial plan, identified as one of the key success factors for projects.

As part of our future work agenda, we plan to extend the capabilities of our simulation approach to handle multi-project resource sharing.

REFERENCES

Arauzo, J. A., J. M. Galán, et al. (2009). "Multi-agent technology for scheduling and control projects in multi-project environments. An Auction based approach." Inteligencia Artificial **42**: 12-20.

Cox, D. R. (1955). "The analysis of non-Markovian stochastic processes by the inclusion of supplementary variables." Proceedings of the Cambridge Philosophical Society **51**(3): 433-441.

Denning, P. J. and R. D. Riehle (2009). "The profession of IT Is software engineering engineering?" Communications of the ACM **52**(3): 24-26.

Herroelen, W. and R. Leus (2005). "Project scheduling under uncertainty: Survey and research potentials." European Journal of Operational Research **165**(2): 289-306.

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

Huang, W., L. Ding, et al. (2009). Project Scheduling Problem for Software Development with Random Fuzzy Activity Duration Times, Springer.

Isensee, C., S. Lazarova-Molnar, et al. (2005). "Combining Proxels and Discrete Phases." Proceedings of ICMSAO.

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

Lazarova-Molnar, S. (2005). The proxel-based method: Formalisation, analysis and applications, Otto-von-Guericke-Universität Magdeburg, Universitätsbibliothek.

Lazarova-Molnar, S. (2005). The Proxel-Based Method: Formalisation, Analysis and Applications. Faculty of Informatics. Magdeburg, University of Magdeburg. **Ph.D.**

Lazarova-Molnar, S. and R. Mizouni (2010). Floating Task: Introducing and Simulating a Higher Degree of Uncertainty in Project Schedules, IEEE.

Lazarova-Molnar, S. and R. Mizouni (2010). "Modeling Human Decision Behaviors for Accurate Prediction of Project Schedule Duration." Enterprise and Organizational Modeling and Simulation: 179-195.

Matta, N. F. and R. N. Ashkenas (2003). "Why good projects fail anyway." Harvard Business Review **81**(9): 109-116.

Sobel, M. J., J. G. Szmerekovsky, et al. (2009). "Scheduling projects with stochastic activity duration to maximize expected net present value." European Journal of Operational Research **198**(3): 697-705.

Stewart, W. J. (1994). Introduction to the Numerical Solution of Markov Chains., Princeton University Press.

AUTHORS BIOGRAPHY

SANJA LAZAROVA-MOLNAR is an Assistant Professor in Computer Science at United Arab Emirates University. She received her Diploma in Computer Science from the "Sts. Cyril and Methodius" University in Macedonia and her M.Sc. in Computational Visualistics and Ph.D. in Computer Science from "Otto-von-Guericke" University of Magdeburg in Germany. Her main research interests are simulation and modeling, with the main focus on their methodology for achievement of more accurate and valid results. Her e-mail is sanja@uaeu.ac.ae.

RABEB MIZOUNI is an assistant professor in the Department Software engineering at Khalifa University, Abu Dhabi, UAE. She got her PhD and her MSc in Electrical and Computer Engineering from Concordia University, Montreal, Canada in 2007 and 2002 respectively. After her graduation, Rabeb joined SAP research labs in Montreal where she acquired experience in the development of mobile applications. Her research interests include modeling and simulation of software requirements to improve the development of robust project schedules, development of mobile applications, and web services. Her email is rabeb.mizouni@kustar.ac.ae.

NADER KESSERWAN got his Master Degree in Computer Science from McGill University. Montreal, Canada. He joined the college of Information Technology as Instructor in 2001. Before joining the academia, he acquired experience in diverse fields of computer science, simulation and real time systems. He spent two years at CAE Company, Canada working on a Flight Simulator, and on testing the integration of its software. Also, he had worked two years on games development for Disney Studio, Canada and USA. He taught programming courses and software engineering at McGill University, Canada. His experience includes team management, system analysis, software development and project management. His email is nkesserwan@uaeu.ac.ae.