Agent-based Modeling and Simulation Framework for Enhanced Project Schedules

Sanja Lazarova-Molnar [a]

[a] *College of Information Technology, United Arab Emirates University, United Arab Emirates*
Email: sanja@uaeu.ac.ae

Abstract: Project schedules are essential elements in project planning. The common problem with them is that the initial project schedules are typically too rigid to reflect the surrounding uncertainties. For this reason we have developed an enhanced project schedule model whose design is simulation-based and incorporates a high degree of uncertainty that typically surrounds project developments. Enhanced project schedules (EPS) are dynamic and allow for pre-calculated beneficial interventions when uncertainties arise. This implies that they feature decision support too. The high degree of uncertainty is incorporated by providing, in addition to the initial project schedule, a set of remedial actions recommendations. Identifying the appropriate set of remedial actions that supports meeting project goals is challenging and carried out through simulation. To enhance the simulation processes that facilitate the design of the EPS and better reflect the uncertainty introduced by the human factor, we propose to use agent-based simulation, for which in this paper we provide the basic framework. The idea is to model tasks, teams and manager as agents with properties and interactions that would more realistically model their dynamic nature. We believe that this will add another dimension to how project schedules are viewed and analyzed. In addition, it will enable more realistic modeling of project schedules and allow for higher degree of dynamics in the generated enhanced project schedules. In this paper we formally define the framework for agent-based modeling and simulation of enhanced project schedules.

Keywords: *Agent-based Modeling and Simulation, Project Schedules, framework*

1. INTRODUCTION

Project schedules are an essential element in project management. There are a lot of research efforts dedicated to design of optimal project schedules, as well as improving the existing project description formalisms. Simulation has always played an important role in project schedule analysis and project management. One of the early efforts is recorded in 1985 (Ahuja and Nandakumar, 1985) where discrete-event simulation is utilized to predict project completion times. Furthermore, in (Lee, 2005) a software, termed Stochastic Project Scheduling Simulation (SPSS) is introduced that also uses simulation to calculate the probability that a project is completed within a given deadline. As of latest, in (Fang and Marle, 2012) a simulation-based risk-network model is presented that should provide decision support in project risk management. To meet the requirement of having more dynamic project schedules to better reflect the real uncertain environment, we have designed the enhanced project schedules model (EPS) that supports the design of more realistic and insightful project schedules with integrated decision support. Each EPS is designed using simulation. So far, we have utilized the proxel-based simulation (Lazarova-Molnar and Mizouni, 2011) and discrete-event simulation (Lazarova-Molnar and Mizouni, 2013). However, to increase the accuracy of our model and enhance the level of realistic features that can be modeled, we have decided to study the agent-based modeling and simulation for the design on enhanced project schedules. This is a novel way of viewing this phenomenon, as we view task and teams as agents. Task as an agent has been observed for agent-based load balancing in grid computing (Wang et al., 2003). In this paper we focus on the conceptual design of such agent-based modeling and simulation for EPS.

2. BACKGROUND WORK

In the following we provide the basis of agent-based modeling and simulation, as well as its use as decision support in a number of management related domains. We further describe our specific application area, i.e. the Enhanced Project Schedule model, along with its formal specification.

2.1. Agent-Based Modeling and Simulation

Agent-based modeling and simulation is gaining in importance and has emerged as a preferred simulation method for certain classes of systems (Macal and North, 2010). This is especially true for systems that include models of behavior, which is true in the case of project schedules. Agent-based models are defined in terms of agents that are described by a set of properties and interact with other agents, as described by simple rules and these interactions further influence agents' behaviors. So far, agent-based modeling has been successfully applied to a number of domains related to management, such as: supply chain management (Labarthe et al., 2007), crisis management (Schoenharl and Madey, 2011), threat management in water distribution systems (Zechman, 2011), management for healthcare (Stainsby et al., 2009), etc. Agent-based modeling is suitable for the above-mentioned domains mainly due to the high degree of uncertainty that each of them exhibits and flexibility of agent-based modeling to reflect it accurately. This also applies to the domain of project management, as there is high degree of human factor involved that needs to be reflected through various behavior models.

According to (Wooldridge and Jennings, 1995), agents are defined as: a) autonomous (able to operate without intervention by humans, and a certain degree of control over its own state), b) reactive (able to perceive an environment in which it is situated and respond to perceived changes), c) proactive (able take the initiative, starting some activity according to internal goals rather than as a reaction to an external stimulus), and d) sociable(able to interact employing some kind of agent communication language). In an agent-based modeling and simulation framework, all of these aspects are formally and represented and described.

2.2. Enhanced Project Schedules

Enhanced project schedules are a novel way of describing project schedules both incorporating project dynamics, and providing guidance to project managers throughout project's implementation. EPS design is accomplished through simulation, which facilitates the selection of an optimal remedial action scenario to complement the initial project schedule, and yield the EPS, as shown in Figure 1.

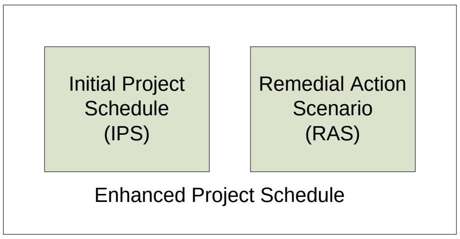
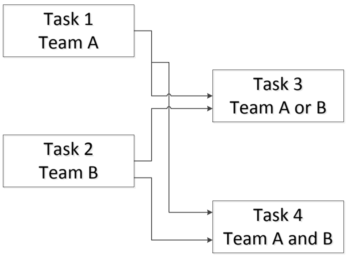

Initial Project Schedule (IPS)
Remedial Action Scenario (RAS)
Enhanced Project Schedule

Figure 1. Enhanced Project Schedule Components



RAS:

1) If duration of task Task2 performed by team B is "very short" then start Task3 by team B.

2) If duration of Task1 is "too long" and it completes "shortly after" team B started to work on Task3, then Task3 is cancelled and both teams start working on Task4.

Figure 2. Illustration of Enhanced Project Schedule (Task Precedence Diagram and Remedial Action Scenario)

In the following we provide an example EPS to demonstrate the novelty in this model. Left side of Figure 2 illustrates an initial project plan shown as tasks precedence diagram. The project schedule consists of four tasks (*Task1*, *Task2*, *Task3*, and *Task4*) assigned to two available teams (*Team A* and *Team B*). The complete Figure 2 (both left and right side elements) illustrates the enhanced project schedule (EPS) model that we propose. While having the same number of tasks and teams, two major features are added:

a) "*floating task*" (*Task 2*), which is a task that can be executed by any of the two teams, albeit its duration is described with different distribution functions (based on teams' expertise), and

b) *remedial action scenario* (RAS), provided on the right-hand side, which is meant to accompany the project schedule as a set of recommendations during its implementation. This RAS suggests that depending on the durations of *Task2* and *Task4*, *Task3* is either cancelled or not. This is an example of a dynamic sequencing, once more information is available.

In the enhanced project schedule the task precedence diagram is used to illustrate the precedence constraints only, without imposing any timing constraints. This implies that the actual starting and ending times of the tasks in the project are uncertain and flexible allowing for more realistic project schedule simulation. Consequently, the model has many scenarios of execution. For instance, and in addition to the scenario described previously and resulting from the classical project schedule, our model may have the following execution scenarios:

1) *Team A* works on *Task1* and *Team B* works on *Task2*,
2) then *Team B* implements *Task 3*,
3) finally the two teams work together to implement *Task4*,

or

1) *Team A* works on *Task1* and *Team B* works on *Task2*.
2) then the two teams work together to implement *Task4*. Implementation of *Task3* is canceled because RAS2 verified.

In our previous work (Lazarova-Molnar and Mizouni, 2010), we were able to successfully model and simulate this type of schedules. We developed an approach to analyze and simulate the effects of uncertainties and remedial actions on duration of projects. To account for resource re-allocations, we have defined the "floating task". This task is typically a task associated with low risk of implementation and could be implemented by a number of teams, albeit with different duration distribution functions expressing their expertise levels. During the project implementation, and based on the team availabilities, one of the specified team will implement the task. In the same work, we showed that modeling *on-the-fly* decisions makes a significant difference in predicting of durations of projects and, consequently, needs to be considered.

RAS are modeled using fuzzy rules. These rules benefit from having a solid theory of straightforward conversion of linguistic expressions into fuzzy membership functions (Zadeh, 1975). The fuzzy description of the remedial action scenarios provides project managers with a degree of flexibility and freedom in their interpretation, thus supporting their own management policies. Project managers can input their own perspectives, values and estimates. As a result, the actual project implementation schedule will be a combination of simulation results and project manager's judgments.

Formal Description of Enhanced Project Schedules

Formally, each project schedule consists of two components: an initial project schedule and a remedial action scenario (RAS), as shown in Figure 2. RAS consists of a set of fuzzy *if-then* production rules. These rules make the project enhanced and thus, the sequencing of tasks - dynamic.

<u>Definition 1 (EPS):</u>

Enhanced Project Schedule (EPS) is described as:

$$EPS = (A, P, O, T, D, W, F, IPS, \Delta)$$

where:

- $A = \{A_1, A_2, \dots, A_n\}$, 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 represented as tuples of two tasks where the completion of the first one is a pre-requirement for the commencement of the second one, e.g. (A_x, A_y) would mean that completing of A_x is a pre-requirement for beginning A_y,
- $O = \{O_1, O_2, \dots, O_r\}$, set of external constraints that are tuples of a conditional statement, and a corresponding action, e.g. $(date > May - 05 - 2011, begin(A_x))$. The conditional statement needs to evaluate to TRUE, so that the action can take place. These serve to introduce constraints that are outside of the project schedule environment.
- $T = \{T_1, T_2, \dots, T_l\}$, 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, each element of W is a triplet of a task, team and distribution function, e.g. $(A_x, T_y, d_z) \in W$ would mean that the task A_x can be performed by team T_y , and the duration in that case would be described by the distribution function d_z,
- $F = \{f_1, f_2, \dots, f_s\}$, set of fuzzy rules that define the optimal remedial action scenario, dependent on the predefined goal(s) of the projects,
- $IPS - Initial\ Project\ Schedule$, initial sequencing of tasks that satisfies the set of precedence constraints provided by P, and
- $\Delta -$ pre-determined deadline of the project.

$P \subseteq A \times A$ is the set of precedence constraints, and $W \subseteq A \times T \times D$ the set of mappings of distribution functions to tasks and teams. Further, the set of tasks A is a union of two disjoint sets: $A = A^c \cup A^n$ $(A^c \cap A^n = \emptyset)$, 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. Another differentiation among tasks is based on the preempt-ability of tasks, i.e. the set of tasks is also partitioned into two sets, based on this feature: $A = A^p \cup A^{np}$ $(A^p \cap A^{np} = \emptyset)$, where A^p is the set of preemptable tasks and A^{np} is the set of non-preemptable tasks. A non-cancelable task cannot be preemptable, as those are vital for the success of the project. The opposite, however, is not true, i.e. a non-preemtable task can also be cancelable. These differentiations are important for the realistic simulation of project schedules.

Each fuzzy rule is 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 enhanced, 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 allow for their modeling, assessment and quantitative evaluation. This makes it straightforward to study the tradeoffs between the various RAS and test for the best possible RAS to balance the 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 IPS. This is the default RAS and it helps in assessing the effects that the proposed RAS has on the original baseline schedule. Once an optimal remedial action scenario is selected, it is associated with the project schedule. This is further demonstrated by a simple example in the following section.

Finally, we define *floating tasks* that are reflection of the flexibility of our approach.

Definition 2 (Floating Task): If a task A_x can be performed by at least two teams, i.e.

$$\exists i, j,\ w_{i,} w_j\ \in W,\ w_i \neq w_j, w_{i_1} = w_{j_1} = A_x,$$

then this task is termed as *floating task.*

3. AGENT-BASED MODELING AND SIMULATION FRAMEWORK FOR ENHANCED PROJECT SCHEDULES

In the following we provide the details of the agent-based modeling and simulation framework for EPS. In particular, we identify the agents and their behaviors.

3.1. Agents

The basic EPS model features three types of agents: manager, tasks and teams. The Initial Project Schedule and the Remedial Action Scenario (i.e. the EPS) belong to the environment of the multi-agent system. Manager is envisioned as an agent that takes decisions based on the Remedial Action Scenario. In the following we detail each type of agent.

Task represents a principal execution unit of a project schedule. It is described by various priorities, constraints, etc. More specifically, each task has the following properties:

a) ID, that represents a unique ID number of a task
b) Description, description of the task
c) Priority, a number that specifies the level of importance of a task (0 – highest priority)
d) Preceding Tasks, a set of task IDs of tasks that need to be completed as a prerequisite to commence the actual task
e) Teams-Durations Distribution Mappings, set of two-element vectors that map teams competent to carry out the task to probability distribution functions that describe durations of tasks when corresponding teams perform them
f) Cancelable, a Boolean value that denotes whether a task can be canceled if necessary or not
g) Interruptible, a Boolean value that denotes whether a task can be interrupted if necessary or not
h) InProgress, a Boolean value that denotes whether a task is being performed
i) Completed, a Boolean value that denotes whether a task has been completed
j) Cancelled, a Boolean value that denotes whether a task has been cancelled

Thus, each task is represented by the tuple:

Task = (ID, Description, Priority, PrecedingTasks, Teams-Durations, Cancelable, Interruptible, InProgress, Completed, Cancelled)

Team is the principal performing unit that needs to be mapped against a task to add value to the project. Each team is described by the following properties:

a) ID, that is a unique ID number of a team
b) Tasks-Durations Distribution Mappings, set of two-element vectors that map tasks that the team is competent to carry out to probability distribution functions that describe durations of corresponding tasks when this team perform
c) Available, a Boolean value that denotes if a team is available or not
d) Completed, a Boolean value that denotes if a team has completed its mission, and, thus, is out of scope
e) Task-InProgress, ID of the task that the team is working on, *null* if team is idle

Thus, each team is represented by the tuple:

Team = (ID, Tasks-Durations, Available, Completed, Task-InProgress)

At our macro level of modeling, we assume that a change in a team structure is reflected by a creating a new team, and successively reflecting this throughout task-agents properties.

Manager is the principal decision unit that is described by parameters extracted from the work described in (Malach-Pines et al., 2009) and filtered based on our model-relevancy:

a) risk factor, (risk taking factor) a number between 0 and 1; assessed using hypothetical situations that require a choice between two options: one involves great risk, the other no risk
b) risk-estimate, a number between 0 and 1 that describes what type of action’s risk factor is estimated as “risky” according to this agent-manager
c) RAS-rules applied, a set of RAS-rules that the manager has applied so far (these, as historical data, could serve to assess managers for future analysis)

Each Remedial Action Scenario comprises of a set of fuzzy rules, and each fuzzy rule is associated with an adventure factor that defines the complexity of the action and the level of risk associated with it. In the following, we define the interactions among agents.

3.2. Environment

The environment of our agent-based model is represented by the entities that affect agents’ behaviors, but cannot be described as agents, i.e. they do not sense and do not carry out any actions. Thus, we define the environment in terms of the Remedial Action Scenario and Initial Project Schedule, i.e. the complete EPS model. This is the data that affects agents’ behaviors in our model.

3.3. Behaviors

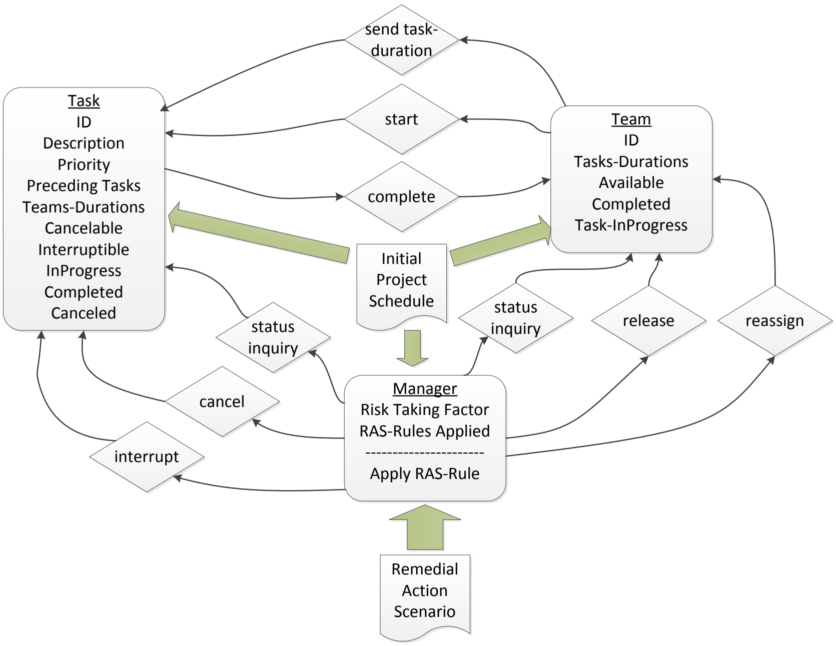


Figure 3. Agent-based modeling and simulation conceptual framework diagram

Specification of agents' behaviors is composed of two elements: a) actions that can either cause change in the environment or in the other agents, and b) mechanisms for selecting appropriate actions based on perceptions and agent's state. In Figure 3, we illustrate the actions that affect the system, and point-out specifically the affected entities. E.g. team-agent can "send task-duration" to a task-agent, and therefore, update the teams-duration property of the task. This can happen when a new team is being introduced, i.e. a change in a team structure has occurred. Additionally, a team can "start" working on a task, and thus change the value of InProgress property of the corresponding task to "true". Due to the space limitation, we only describe on mechanism in detail, i.e. manager's behavior, as shown in Algorithm 1.

Algorithm 1. Mechanism of manager's behavior

```
for-each fuzzy rule RAS$_i$ in RAS
   {if (RANDOM1 < eval(RAS$_i$.if-part, t) and
      ((RAS$_i$.adventure-factor <= manager.risk-estimate) or (RANDOM2 > manager.risk-factor))
               then perform RAS$_i$.then-part }
```

To illustrate this mechanism we use the following example fuzzy rule: *if Task A is taking "too long" ⇒ interrupt Task A*. We evaluate it at a time *t*, which implies the need to evaluate the fuzzy function "too long" at this time too by a number F between 0 and 1. We generate a uniform random number R and compare the value of fuzzy function. If $R < F$ then the if-part is evaluated to true. Also if the RAS rule risk factor is less than the one of the manager, then he/she would carry out the action. However, if this is not the case then we generate a random number to reflect the probability that he/she would carry out an action assessed as risky. If manager's risk factor is 0.7, it implies that he is prepared to take risky actions 70% of the time. A risky action is an action with higher risk factor that the "risk estimate" property of the manager.

In Table 1, a summary of all communication messages is presented, along with the sending and receiving agents, as well as message description and its associated action.

Table 1. Communication messages used in the framework (Mng stands for manager)

Message Type	From	To	Description	Action
Send task duration	Team	Task	New team sends its duration distribution function to the task	Add new element to the taks's teams-durations set.
Start task	Team	Task	Team starts working on a task	Task's InProgress set to true and team's Available to false, and Task-InProgress to tasks's ID.
Task Completed	Task	Team	Task is completed	Task's InProgress set to false, Completed to true, and team's Available to true.
Task Cancelation	Mng	Task	Cancel task	Task's Canceled set to true.
Task Interruption	Mng	Task	Interrupt task in progress	Tasks Canceled set to true.
Team Release	Mng	Team	Release team from working on a	Team's Available set to true.
Team Reassign	Mng	Team	Change the task that the team is working on	Replace team's Task-InProgress with the ID of the new task.
Task Status Inquiry	Mng	Task	Manager requests a status update from a task	Generate a status return message.
Team Status Inquiry	Mng	Team	Manager requests a status update from a team	Generate a status return message.

4. DISCUSSION AND CONCLUSIONS

We have presented a conceptual framework to support the agent-based modeling and simulation of enhanced project schedules. The aim of our work is to enhance our project schedule model to better reflect the various uncertainties that can arise due to the human factor, as well as other environmental factors. We believe that the agent-based model is more flexible and better suited for the simulation of environments that involve people and their behaviors. The goal of the simulation is design an optimal RAS with respect to the characteristics of the manager and initial project schedule. In addition to this, another interesting aspect is treatment of a task as an agent, which we see as a potential for a task to include decision support within itself, as well as feature some intelligence through various heuristics, memory and historical data analysis. We intend to extend this work with suitable graphical representation and interpretation of EPS models and implement it as part of a comprehensive decision support system.

REFERENCES

Ahuja, H.N.; Nandakumar, V. Simulation model to forecast project completion time. Journal of construction engineering and management. 1985;111(4):325-342.

Lee, D.-E. Probability of project completion using stochastic project scheduling simulation. Journal of construction engineering and management. 2005;131(3):310-318.

Fang, C.; Marle, F. A simulation-based risk network model for decision support in project risk management. Decision Support Systems. 2012;52(3):635-644.

Lazarova-Molnar, S.; Mizouni, R. A simulation-based approach to enhancing project schedules by the inclusion of remedial action scenarios. Proceedings of the Winter Simulation Conference: Winter Simulation Conference; 2011. p. 761-772.

Lazarova-Molnar, S.; Mizouni, R. Discrete-Event Simulation for Design of Enhanced Project Schedules. Summer Computer Simulation Conference. Toronto, Canada; 2013.

Wang, Y.; Liu, J.; Jin, X. Modeling agent-based load balancing with time delays. Intelligent Agent Technology, 2003 IAT 2003 IEEE/WIC International Conference on: IEEE; 2003. p. 189-195.

Macal, C.M.; North, M.J. Tutorial on agent-based modelling and simulation. Journal of Simulation. 2010;4(3):151-162.

Labarthe, O.; Espinasse, B.; Ferrarini, A.; Montreuil, B. Toward a methodological framework for agent-based modelling and simulation of supply chains in a mass customization context. Simulation Modelling Practice and Theory. 2007;15(2):113-136.

Schoenharl, T.; Madey, G. Design and implementation of an agent-based simulation for emergency response and crisis management. Journal of Algorithms & Computational Technology. 2011;5(4):601-622.

Zechman, E.M. Agent-Based Modeling to Simulate Contamination Events and Evaluate Threat Management Strategies in Water Distribution Systems. Risk Analysis. 2011;31(5):758-772.

Stainsby, H.; Taboada, M.; Luque, E. Towards an agent-based simulation of hospital emergency departments. Services Computing, 2009 SCC'09 IEEE International Conference on: IEEE; 2009. p. 536-539.

Wooldridge, M.; Jennings, N.R. Intelligent agents: Theory and practice. Knowledge engineering review. 1995;10(2):115-152.

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

Zadeh, L.A. The concept of a linguistic variable and its application to approximate reasoning--I* 1. Information sciences. 1975;8(3):199-249.

Malach-Pines, A.; Dvir, D.; Sadeh, A. Project manager-project (PM-P) fit and project success. International Journal of Operations & Production Management. 2009;29(3):268-291.