Advanced Engineering Technology and Application
An International Journal

http://dx.doi.org/10.18576/aeta/050201

Hybrid Tree for Scenario Modeling of Project Management

Zhassan Baipakbayev *

Kazakh - British Technical University, Almaty, Kazakhstan

Received: 20 Feb. 2016, Revised: 7 Apr. 2016, Accepted: 8 Apr. 2016
Published online: 1 May 2016

Abstract: Most often projects fail due to the management problems. Some management problems do not happen because of bad managers, but because of risks, which cannot be prevented mentally. Therefore, the project managers, especially who work on big projects, need proactive tools to prevent all possible ambiguous risks. Classical project management solutions try to solve the problem by the probabilistic approach. But this kind of solution does not analyse the all possible scenarios. Therefore experts do not rely on them. In this paper we suggest a theory, which gives an opportunity to analyse all possible scenarios as much as possible.

Keywords: Scenario Modeling, Project Management, Resource Management, Scenario Trees

1 Introduction

Problem of in detail planning is still unsolved for complex projects [1]. Classical project management approach has not changed from the early 1960th, and they indeed have been the only theory, which stay on the core of the project management. All the other theories were not implemented to practical project management tools. At the present time, majority of the project management tools are just like organizers in general. For instance asana.com, todois.com etc. Only few of them have functionalities, which allow to build the PERT diagrams and CPM charts.

So far the project management process have been considered in terms of activities or milestones. Such an approach has several advantages, like identification of the critical path activities, estimation of the project duration time and etc. However it cannot prevent the project from failures and back-side scenarios, which often happen to different kind of activities. Despite the risk evaluation techniques estimate the risks which could happen, they do not provide the way to treat the risks before they appear.

Risk and uncertainty arise from measurement errors and from the underlying variability of complex, natural, social, and economic situations. If the analyst is uncertain because of imperfect data or crude analytical tools, the plan is subject to measurement errors [2]. To minimize the measurement errors there is need to create a common framework for all project management cases. One of the most impressive theories is the Systems Approach by Harold R. Kerzner, that has led us to a different concept of project management [3]. The latter is successfully adopted for construction projects [4]. However, other types of projects like software, logistics etc. still need concrete managemental treatments.

In this paper, we present a theory which extends the classical approach a little bit, in terms of scenarios. We know, that treating each risky situation leads us to the new scenario of the project workflow. Scenarios intuitively lead us to scenario tree contruction. The similar approach was suggested for power management problems [5]. If we imagine a little bit, due to the possible scenarios, whole project management process behave like a game. Based on that, we make an analogy with a game theory algorithms, which are also based on management of different scenarios of the game process. We consider popular algorithms like alpha-beta pruning, minimax, then adapt them to the project management.

As a result we can obtain the new way to deal with risks in the project management, which further pretends to be the core of future proactive project management theory.

* Corresponding author e-mail: zhasdu@gmail.com

© 2016 NSP
Natural Sciences Publishing Cor.

2 Strategy of computer bots

Garry Kasparov, who had been the best chess gamer, lost to the Deep Blue IBM computer in 1996. Then, people wondered at the result. In fact, there is nothing to wonder, because a computer memory can keep almost all the possible states of a game and make the most beneficial decision. The most popular algorithm the computer bots play according to, is the Minimax. Let us get inside into the basics of the Minimax.

For ease of imagination, let us consider two player games. For two-player games, the Minimax algorithm is such a tactic, which uses the fact, that the two players are working towards opposite goals to make predictions about which future states will be reached as the game progress, and then proceeds, accordingly to optimize its chance of victory [6]. The theory behind minimax is that the algorithm's opponent will be trying to minimize whatever value the algorithm is trying to maximize (hence, "minimax") [6]. Thus, the computer should make the move which leaves its opponent capable of doing the least damage [6]. In the above example (Figure 1), assuming normally alternating turns, if the computer has the choice at level A, its opponent will have the choice at B, and the computer will again have the choice at C [6]. Since the computer is trying to maximize its score, it can know ahead of time what it would choose should any given C node be reached [6]. C1 thus effectively has a score of 5, C2, 11, C3, 8, and so on [6]. When the opponent has a choice to make at B, however, they will choose the path that leads to the C node with the lowest effective score [6]. Thus, the score of each B node can be thought to be the minimum of the effective scores of the C-nodes it leads to [6]. For example, B1's score would be 5 (the minimum of 5, 11, and 8, as calculated above). B2 and B3 can be calculated in a similar fashion [6]. Finally, we are back to the current turn [6]. The computer now knows what will come of choosing B1, B2, or B3; and, though the actual endgame is many turns away, it will choose the maximum of those three for the best possible result [6]. Note that, if the opponent does not behave as predicted, the calculation can simply be re-run, taking the current state as the starting node, and a result as good (or better) than what was predicted will still be achieved [6].

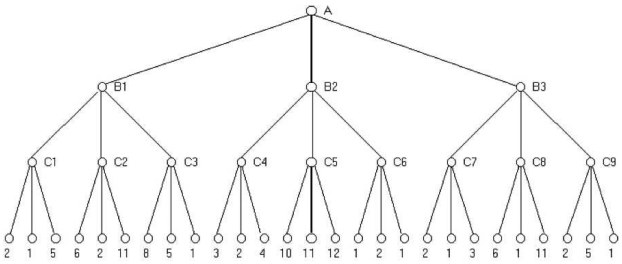

Fig. 1: Minimax Tree. Copied from web-site of the Stanford University

The simple minimax algorithm described above, makes full traversal and considers all the states starting from the root to leaves. However, if a game has very large number of possible scenarios, then it takes too long time to analyse all the states. Indeed, in real life mostly we are to model huge number of scenarios. Therefore, there is need for optimization.

Well-known optimization for the algorithm is alpha-beta pruning. Alpha-Beta pruning method introduces two types of the tree nodes, so called, minimizers and maximizers. Minimizer nodes take the minimum of the possible choices, and maximizers take the maximum of the possible choices. In a game they correspond to the players and its opponents decision nodes. To understand the alpha-beta pruning, let us imagine that we are playing a game, where we want to maximize points, whereas the opponent tries to minimize it. Let us assume that we are at state A (Figure 2). By the way, the maximizers are presented as base-down triangles and minimizers as the reversed ones in the figure. It is shown that each node stores some number. The numbers stand for the choices would be made at corresponding state. So, we see that at node A we have two choices, either to go left or right. Here it is straight seen which points would be equal to in both choices. As a result we can skip the whole left part of the tree at once. In such a way, we do the same pruning operation over the whole tree.

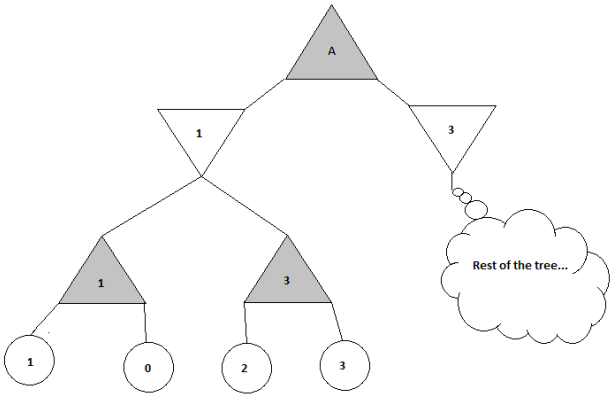

Fig. 2: Alpha-Beta Pruning

In turn, the project management can be treated as a two player game where the project owners are the players who want to increase the outcomes, and there is an opponent who plays in order to decrease the outcomes. In addition, this method is quite convinient for software process simulations [7].

© 2016 NSP
Natural Sciences Publishing Cor.

3 Scenario from resource management perspective

Usually, who deal with project management are used to understand activities or possible decisions under scenarios. Here we will consider scenarios from resource perspective.

In general we can divide project into activities and resources. Here activities are the actions done upon the resources by other resources. Activities needed to be done describe the states of a project. In turn the states define the scenarios of the project workflow. Whereas required activities at any state depend on resources state. Consequently, we conclude that scenarios of the project workflow are defined by states of resources.

4 Resource states

States of a resource are, in some meaning, subjected to the resource reliability. There are different types of resources like, machinery, men power, software etc. Correspondingly, states of the resources of different types also differ. So far, there was unclarity in defining software resource states, therefore they can be estimated by the Weighted Isotropic Laplacian Approach [8]. Concerning other types of the resources, it is the matter of rigorous analysis.

Any resource is at some state, at any moment of time and at any point of the space. For example, let us consider a laptop. Lap top is either turned off, working, asleep, hibernated or crashed. These are the general states of laptop, which is a resource. Exactly each of those general states can be divided into sub-states. For instance, working laptop can be either virused or healthy or without OS or etc. Similarly, any resource can be referred to some of the predefined states of it.

Definition of resource states is more complex than it seem to be. States of resources are individual for each resource. Therefore it is possible to define very large amount of resource states. Our aim in this paragraph is to simplify this process.

In order to reduce number of possible states, we consider only those states which may occur at the given period of time. To estimate time when the resources may appear at a certain state let us remind classical CPM(Critical Path Method). The Critical Path Method or Critical Path Analysis, is a mathematically based algorithm for scheduling a set of project activities [9]. It is an important tool for effective project management [9]. Commonly used with all forms of projects, including construction, software development, research projects, product development, engineering, and plant maintenance, among others [9]. Any project with interdependent activities can apply this method of scheduling [9]. To construct CPM graph it is necessary to define all the activities during the project workflow. We know that each activity uses certain resources. The critical resources are the resources used by the most of activities and additionally resources which are used by critical path activities.

Because CPM chart and PERT diagrams are well known techniques, we do not focus on details, and proceed to the following example (Figure 3). In the example above we

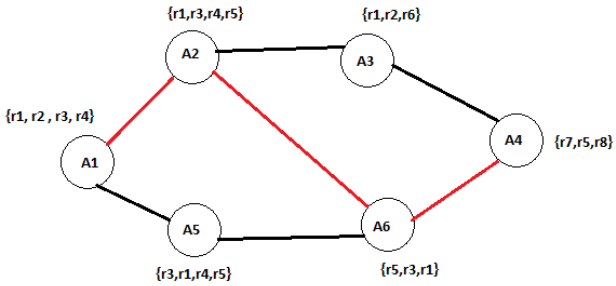

Fig. 3: Critical Path Method (CPM)

see CPM chart consisting of 6 activities. Each activity is based on certain set of resources. Activities A1, A2, A6 and A4 are Critical Path Activities. Goal of the CPM was to determine that path, but this information is not so important for us. Our aim is to identify resources utilization time. For instance, resource r5 is utilized only in the time range from the moment A1 finishes to A4 finishes, resource r4 is utilized in time range between A1 starts up to A2 finishes etc. According to that information, while constructing our tree we will not consider r5 in the period of activity A1, and we will not consider states of r1 in the period of A4 and so on. In turn the activity duration is subjected to Resource-Constrained Project Scheduling [10].

5 Scenarios as states of resources

Each state of a project is described by the set of resources. Resources states differ by the set of their current states. Let us go to the example from the previous paragraph. The project starts with activity A1, and it is based on the set of resources r1,r2,r3,r4. Let us define state as a function of resource and time. Then assume that,

$$s(r1,t) \in \{1A,1B,1C\}; \quad (1)$$

$$s(r2,t) \in \{2A,2B,2C,2D\}; \quad (2)$$

$$s(r1,t) \in \{3A,3B\}; \quad (3)$$

$$s(r4,t) \in \{4A,4B,4C,4D\}. \quad (4)$$

© 2016 NSP
Natural Sciences Publishing Cor.

Then, we can conclude that there are

$$N = |D(s(r_1,t))| \times |D(s(r_2,t))| \times |D(s(r_3,t))| \times |D(s(r_4,t))| \quad (5)$$

possible states of the resources, consequently the same number of states of the project. Each unique state is equivalent to separate scenario. For instance, one of the possible states of the resources from the current example is (1A, 2B, 3A, 4C). In general the number of scenarios from resources perspective is equal to

$$N = \prod_{i=1}^{n} |D(s(r_i,t))|, n \in \mathbb{N}. \quad (6)$$

Where n is the number of resources available in the current activity.

6 Resource state tree

In this paragraph we define n-ary tree, based on the resource states described in the previous one. Let us remind that the number of resources involved into the process change by periods of time. Therefore the number of states of the resources involved into the process are also restricted by time periods.

In a certain time period, the transitions between states of resources are a special case of the Markov chains [11]. Because we need to consider each state of resources equally probable to happen at any time, the size of tree is limited by time and cost constraints. To implement that we need to declare two parameters time and cost at the root of three. Then, dynamically keep altering them during the tree traverse while they are positive.

For example, let us consider a single-activity fruits transfer project. The projects goal is to transfer fruits from point A to point B for 2000 km. The resources involved in this project are truck(¡v¿=50km/h), driver, fruits. Let us define states of the resources,

$$s(truck,t) \in \{crashed, wheelbroken, no\,fuel, ok\}, \quad (7)$$

$$s(driver,t) \in \{invalid, ok\}, \quad (8)$$

$$s(fruits,t) \in \{soured, ok\}. \quad (9)$$

Suppose that, we are given 2 days of deadline, 1000$. Also we know that driver should eat each 5 hours, and sleep for 8 hours, lunches cost 20$ in average, truck needs to be refilled with fuel after 200 km and one refillment costs 30$. Other parameters are shown in the table below (Table 1). In the table we see MBTF, LTTR and COR parameters for each state different from ok. Note that we do not consider MTTR (Mean Time Between Failures) but LTTR, because we need to determine worst case effects.

For demonstration of the example above, we have

Table 1: (LTTR - Longest Time To Repair, MTBF - Mean Time Between Failures, COR - Cost Of Repair)

States	LTTR,hours	MTBF,hours	COR,$
Crashed	12	8760	980
No fuel	12	4	30
Wheel broken	12	720	0
Invalid	1	5	20
Soured	24	48	10000

simulated the process using simple demo program, which is publicly accessible at GitHub repository [12]. Then we have obtained tree of states from the root of the tree (dummy initial state) till the leaves, where leaves are either logical end of the project or crash states, where either time or cost is negative. Let us look at the fragment of the tree with crash states (Figure 4). In the figure it is

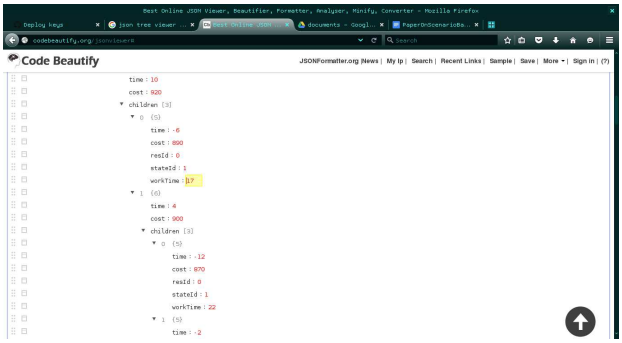

Fig. 4: Resource state tree JSON

shown that after 17 hours of travel resource id 0 (truck) may go to state id 1 (No fuel), and it may cause the project failure due to long time delay to repair the resource (look at the child where time is -6). On the next branch of the tree we can see that the same problem may cause again the project failure.

7 Decision tree vs Resource state tree

In the previous paragraph, we have explained resource state tree with the example. Here we will compare well-known decision tree with our resource state tree, in order to clarify their pros and cons.

Decision trees are based on decisions could be made by agents or users and their consequences. The main advantage of this kind of tree is the detection of optimal decision among possible ones. However, basically in project management the scenarios are predicted by resources. Because of that fact, decision trees are not able to provide deep analysis in that area.

Resource state trees give us great opportunity to analyse all the possible scenarios in project management,

© 2016 NSP
Natural Sciences Publishing Cor.

moreover they could be very helpful in predicting, so called, black-swans in the project workflow. Unfortunately pure resource state trees can not be applied iteratively during the whole activity lifetime, because it ends up with crash states. Another big disadvantage of the resource state tree is that it focuses only upon worst case states, whereas decision tree would give us chance to avoid many of them.

To sum up, neither using pure decision tree nor pure resource state tree is effective in scenario modeling of project management. However they indeed complement each other. Therefore good strategy is to combine them, and produce hybrid scenario tree.

8 Hybrid tree

We call hybrid tree the combination of the decision tree and the resource state tree. This approach gives us opportunity to iteratively go through the whole activity lifetime, being able not only to detect dangerous scenarios but also to handle them when it is possible.

Let us remind that in resource state tree decisions at each state was made at once according to the worst case and there was only one type of nodes. In the hybrid tree we define two types of nodes. Let us call them RSN (Resource State Node) and DSN (Decision State Node).

The RSNs describe all the possible states of resources at each stage, just like in resource state trees. The only difference between RSNs and the nodes of the resource state tree is that no decisions are made immediately at RSNs. DSNs usage is quite intuitive. They describe states of the resources after certain decisions. As a result we can handle each state differently instead of judging by worst cases only. In the figure above (Figure 5) shown a hybrid

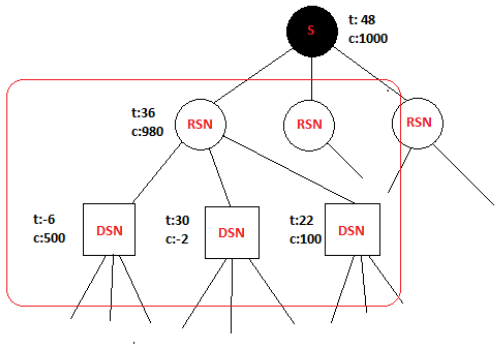

Fig. 5: Hybrid tree

three with two dynamic parameters t (time) and c (cost) which are initially equal to 48 and 1000 correspondingly (example from paragraph 3.1). We see that after 12 units of time the resources become at state with t = 36 and c= 980. Also, we that no decisions made immediately at that RSN. The RSN has three DSN children. Then it is demonstrated that at each DSN outcomes are also different. At the left-most DSN time goes negative which crashes the project, and at the middle DSN cost goes negative, which crashes the project also. In turn the right most DSN ends up with positive outcomes, which give the scenario grow further. This example shows us the advantage of hybrid tree in compare to pure resource state tree, where the same scenario ended up with the only crash state.

In general the hybrid tree looks like a minimax tree, where RSNs minimize the outcomes, whereas DSNs aims maximizing the outcomes. It helps us to find the most stable scenario among possible ones.

9 Conclusion

In this paper we have explained a new approach for scenario modeling of project management. We have analysed the weaknesses and advantages of the classical approaches and tried to bind them in a new theoretical concept. The concept is the analogy of famous gaming algorithm called MiniMax, mapped to the project management. During the research we have detected that different scenarios come from the project resources, therefore our concept is based on the analysis of the resources.

Initially, we have introduced resource state tree, which is a new type of scenario trees. In that part of the paper we have analysed main advantages and disadvantages of our concept in compare to existing methods, and came up with hybrid model for scenario modeling of project management. Still remains the problem of identification of all the possible states of resources, but there exists a bright idea to come up with it in the future [13].

This model needs further improvements and additional tests. However, it pretends to be the base for the integrative scenario modeling framework for project management in the future.

References

[1] T M Williams, The need for new paradigms for complex projects,*International Journal of Project Management Vol. 17, No. 5, pp. 269273*, 1999

[2] Y.Y.Haimes, Risk Modeling Assessment And Management, 2nd Edition, 2004

[3] Harold R, Kerzner, Project Management. A Systems Approach to Planning, Scheduling and Controlling,2009

[4] Anthony Walker, Project Management In Construction, *Wiley conference*, 27.08.2007

[5] Nicole Grwe-Kuska, Holger Heitsch and Werner Rmisch, Scenario Reduction and Scenario Tree Construction for Power Management Problems, *IEEE Bologna Power Tech Conference*, 23-26.06.2003

© 2016 NSP
Natural Sciences Publishing Cor.

[6] Strategies And Tactics For Intelligent Search.Algorithms-MiniMax, *Educational web-site of Stanford Univeristy web.stanford.edu*, 23.02.2016
[7] Marc I, Kellner * , Raymond J, Madachy , David M, Raffo, Software Process Simulation Modeling:Why? What? How?,*Journal of Systems and Software*, 15.04.1999
[8] Chandra Mouli Venkata Srinivas Akana, C. Divakar, Ch. Satyanarayana, "Weighted Isotropic Laplacian Approach for Software Reliability Estimation for a Growth Model",*Advanced Engineering Technology and Application Vol.4, No 1, pp. 7-10*, January 2015
[9] Jesse Santiago, Desirae Magallon,Critical Path Method,*CEE 320 VDC SEMINAR*, 04.02.2009
[10] Daniyar Artykov, Lyazzat Atymtayeva, "A Fuzzy Linear Programming Approach for Resource-Constrained Project Scheduling",*Advanced Engineering Technology and Application Vol.3, pp. 47-52*, May 2013
[11] James Norris, Markov Chains,*University Of Cambridge*, 21.03.2016
[12] Zh.Baipakbayev, "Scenario Modeling Of Project Management", *GitHub Repository zhassanbey/ScenarioModelingOfProjectManagement*,2016
[13] Mrcio de Oliveira Barros,Cludia Maria Lima Werner,Guilherme Horta Travassos, Scenario Oriented Project Management Knowledge Reuse within a Risk Analysis,*68511 - CEP 21945-970 - Rio de Janeiro RJ*, 2011

Zhassan Baipakbayev received the bachelor degree in Information Systems at Suleyman Demirel University in Almaty. His main research interests are: modeling of resource management process, big data, machine learning, algorithms.

© 2016 NSP
Natural Sciences Publishing Cor.