Action Driven Decision Modeling Framework towards Formulating Software Project Management Tacit Knowledge

Ali Tizkar Sadabadi, Nazri Kama
Department of Software Engineering, Advanced Informatics School, Universiti Teknologi Malaysia,
Kuala Lumpur, 54100, Malaysia
Email: al_tz2@yahoo.com, nazrikama@ic.utm.my

***Abstract*—Managing the tacit knowledge in organizations raises substantial challenges in regards of the associated processes. In software project management decision making has the critical role in this scenario since it defines the manager's responsibilities and stems from the various sources linked to the process. With the respect of tacit knowledge, the decision making constructs the essential foundation and thereby it needs a reliable framework for modeling of the decision structure. In this paper a conceptual multi-method simulation based framework will be introduced in a modality to cover multiple levels of the decision structure over software project management process. The methods used are integrated towards a multi-method simulation model whereas each of these methods exclusively realizes distinct aspect of software project management. The framework evolves the manner of decision making by a paradigm which establishes the foundation for a tactical level understanding and decision support for practitioners. At the results section an optimal policy for the framework will be presented.**

***Index Terms*—software project management, action driven decision modeling, tacit knowledge, decision support paradigm**

I. INTRODUCTION

Software project management (SPM) is basically defined by the ability of decision making. It is the responsibility of managers to design this process and optimize it to minimize costs and maximize production. Decision makings are based on resources and constraints which are planned for the target project and the plan could change hardly in line with new requirements during project progress. But what could be done accordingly to confront changes which are contingent in every project, is to define an optimal plan with effective decisions.

Decision making is a cognitive process resulting in the selection of a course of action among several alternative scenarios. This process finally leads the decision maker to take an action or make a choice [1]. Thereby it is an ability based on experience and knowledge that enables a leader of a process to succeed. The nature of software projects on the other side add other complexity in which development process has intangibility that makes it difficult for managers to design a suitable strategy for decision making.

Corresponding author email: al_tz2@yahoo.com.

SPM requires special mindset to make practitioners be able to conduct management process in an effective and efficient manner. These mindsets are from any point of view considered as high level experience and management capabilities, since it originates from a complex process and organizational understanding [2]. Therefore Management of this knowledge requires special strategies for an effective knowledge management.

It is evident that an effective approach for modeling of the decision making process and redefining the decision structure over SPM is necessary. This model should be able to deal with high level of intangibility and continuous change requests within the software project.

In this paper a framework which constructs the basis for modeling of decision making process over SPM will be introduced. This framework incorporates knowledge management discipline and simulation methods as well as the methodology that is supported by SPM discipline. In the framework new concepts will be introduced and the possibility of distinct levels of view over SPM would be realized.

II. RELATED LTERATURE

DSS (Decision support systems) are generally implemented as expert systems into SPM process. The frameworks have been introduced in the literature of Software Project Management Decision Support (SPMDS). With study in the respective literature, scholars [3-10] stated the improvement of decision making by implementation of expert systems in SPM. As stated by Antony and Santhanam [11] with supporting decision-making ability, the use of knowledge-base systems could implicitly improve the learning process as a stimulus. Olteanu [8] addresses the necessity of implementation of DSS in project management to identify all opportunities for improvement of decision value and lowering the production cost. Janczura and Golinska [9] defines the DSS implementation as an appropriate criteria for choosing a model for software development life cycle. At the end the author mentions that selecting an appropriate software development life cycle model is a complex and a challenging task, which requires not only broad theoretical knowledge, but also

doi:10.4304/jsw.9.9.2474-2483

consultation with experienced expert managers. Therefore, the computer application presented should be perceived as the first step towards building a system that could be applied in practice. Besir and Birant [12] stated the DSS use in SPM could avoid the possible erroneous results and help the companies to perform the managing and planning functions easier. As Yang and Wang [13] acclaimed, Project management is an experience-driven and knowledge-centralized activity. Therefore, project managers require some assistance to reduce the uncertainty at the early stage of constructing project plans. Authors applied cased–based reasoning technique for formulating the project requirements.

Other scholars have implemented simulation technique as a method for modeling software engineering process. That is significant to mention there are two types of efforts in the literature focused on SPM modeling. Those that in the literature, named software development process learning improvement (SDPLI) but implicitly they have tried to digest topics of SPM [14-18]. And others that explicitly addressed and focused on SPM experience acquisition solution [19-23]. As noticed in the literature of SDPLI, approaches for improvement of learning scheme for software development process mainly paid attention to provide a set of facilities from models to applications to support learning process of SPM.

The summary of existing works with implementation of simulation and DSS in regards to SPM knowledge acquisition is shown in table I. Thereby the legends on table 1 are described as follow:

Based on the level of understanding over SPM process as E-D: explicit direct view, O: operational, M: managerial, T: tactical, S: strategic and the concepts that are in this paper introduced based on decision support paradigm as DS: decision support, In-P: in-process decision support, Off-P: off-process decision support, N: not considered, Y: considered, P: partially considered. A brief explanation for the levels of view, according to Targowski [24] categorization of organizational and managerial understanding (tacit knowledge) basic, whole, global and universal mind. In this paper they are specified as (1) operational, (2) managerial, (3) tactical and (4) strategic levels. Accordingly in this paper two terms are specified as: The "in-process decision support" which resolves short-term and real-time decision issues and is assumed on the tactical level over SPM process. The "off-process decision support" resolves long-term and past decision issues and is assumed on managerial level over SPM process. Also in the discipline coverage column in table I, SPMDS stands for software project management decision support and SPMLI for software project management learning improvement.

III. Framework Composition

A. Simulation Model

Simulation model is accountable to provide a basis for animation of SPM process. This operability is conducted through the implementation of multi-method simulation technique. The simulation techniques which are applied to this framework are discrete event simulation (DES), system dynamics (SD) and partially observable Markov decision process (POMDP). Therefore these methods will be used to model exhaustive simulation logic and engine.

Discrete Event System Specification (DEVS) [25] atomic formalism allows to develop a DES based system. An atomic DEVS model is defined as a 7-tuple: $M = < X, Y, S, ta, \delta\text{ext}, \delta\text{int}, \lambda >$.

Accordingly simulation model specification according to atomic DEVS would be:

Set of input events $\mathrm{X} = \{\text{useraction}\}$

Set of output events

$Y = \{\text{message}, \text{endofphase}, \text{startsimulation}, \text{endsimulation}\}$

Set of sequential states

$$S = \left\{ \begin{matrix} (\mathrm{d}, \mathrm{a}) |\ \mathrm{d} \in \left\{ \begin{matrix} \text{start}, \text{phase1}, \text{phase2}, \\ \text{phase3}, \text{phase4}, \text{end} \end{matrix} \right\}, \\ \mathrm{a} \in ((0, \text{lenghtofphase}] \cap \mathrm{T}\infty) \end{matrix} \right\}$$

TABLE I.

Characteristics of Existing Works with Implementation of Simulation and DSS in SDPEI

Name	Reference	Discipline Coverage	Explicit	Tacit				DS	
			E-D	O	M	T	S	In-P	Off-P
AMEISE	Bollin et al., 2011	SPMDS	Y	Y	Y	N	Y	Y	N
PMA	Dufner et al., 1999	SPMDS	Y	N	Y	N	N	N	Y
DSS for SR ESS	Rus and Collofello, 1999	SPMDS	P	Y	Y	N	Y	Y	Y
DSS for SPM	Donzelli, 2006	SPMDS	N	N	P	N	N	N	Y
SRNMDSPRM	Fang and Marle 2012	SPMDS	Y	Y	P	N	N	N	Y
EPECCS	Cho 2006	SPMDS	N	Y	P	N	N	N	P
AKBS	Antony and Santhanam, 2007	SPMDS	P	N	P	N	N	N	P
SimSE	Navarro, 2006	SPMLI	Y	Y	N	N	P	N	N
SESAM	Drappa and Ludewig, 2000	SPMLI	Y	Y	N	N	P	N	N
OSS	Sharp and Hall, 2000	SPMLI	Y	P	N	N	N	N	N
PMT	Davidovitch et al., 2006	SPMLI	Y	Y	N	N	N	N	N

Where phase1,…, phase4 are software development life cycle phases considered, $T^{\infty} = [0,\infty]$,

$$\text{lenghtofphase} = \begin{Bmatrix} \text{lenghtofphase1}, \text{lenghtofphase2}, \\ \text{lenghtofphase3}, \text{lenghtofphase4} \end{Bmatrix}$$

is the elapsed time during the phase. This variable is dependent on predetermined value of scheduled time for the phase (based on input variables before the start of simulation) and decisions the performer of simulation will make during progress of the phase. In this simulation it is supposed that "lenghtofphase" never will be zero. For a, the reason we accounted zero for the range (in $T\infty$ range) is just because of $\{\text{start}, \text{end}\}$ states that their lifespan are considered$\approx \varepsilon$.

c= the time value of loading simulation, d= the time value of ending simulation and showing results as stated: $c = d = \varepsilon$.

Initial state $S_0 = \{\text{start}, \text{c}\}$

Time advance function ta(s)=a for all s $\in$ S

External transition function

$$\delta\text{ext} = \begin{Bmatrix} (\text{phase1}, T\infty), (\text{phase2}, T\infty), (\text{phase3}, T\infty), \\ (\text{phase4}, T\infty), (\text{end}, \text{d}) \end{Bmatrix}$$

Internal transition function

$$\delta\text{int} = \begin{Bmatrix} (\text{start}, \text{c}), (\text{phase1}, \text{lenghtofphase1}), \\ (\text{phase2}, \text{lenghtofphase2}), \\ (\text{phase3}, \text{lenghtofphase3}), \\ (\text{phase4}, \text{lenghtofphase4}), (\text{end}, \text{d}) \end{Bmatrix}$$

Output function

$$\lambda = \begin{Bmatrix} \text{startsimulation}, \text{message}, \\ \text{endofphase}, \text{endsimulation} \end{Bmatrix}$$

SD is an approach to understanding the behavior of complex systems over time. It deals with internal feedback loops and time delays that affect the behavior of the entire system [26]. There mainly two topics in SD: (a) Causal loop diagrams, is a simple map of a system with all its constituent components and their interactions. By capturing interactions and consequently the feedback loops, a causal loop diagram reveals the structure of a system. By understanding the structure of a system, it becomes possible to ascertain system's behavior over a certain time period. (b) Stock and flow diagrams, to perform a more detailed quantitative analysis, a causal loop diagram is transformed to a stock and flow diagram.

A POMDP models an agent decision process in a Markov Decision Process, but the agent cannot directly observe the underlying state. Instead, it must maintain a probability distribution over the set of possible states, based on a set of observations and observation probabilities, and the underlying Markov Decision Process [27]. An exact solution to a POMDP yields the optimal action for each possible belief over the world states. The optimal action maximizes the expected reward of the agent over a possibly infinite horizon. Briefly a POMDP consists of 6 elements plus the belief state condition; set of states, actions, observations, state conditional transition probability function, conditional observation probability function and reward function.

1) Simulation methods and their correspondence to the level of views

Each method projects specific level of perspective over simulation process. They operate at different level of abstraction and comprises of distinct elements. DES is the basis for constructing the simulation operability. SD which entails the highest level of simulation perspective that provides strategic view from the system behavior. Yet the multi-method simulation approach is not coherent and there is a gap between these two levels. POMDP fills out this gap and provides tactical view level of process. This level is as much significant as, on one side to coordinate the two different techniques of DES and DS and on the other side to adapt the continuous technique, of SD with the discrete one, of DES. The simulation methods and their correspondence to level of operation from each distinct perspective are illustrated in figure 1.

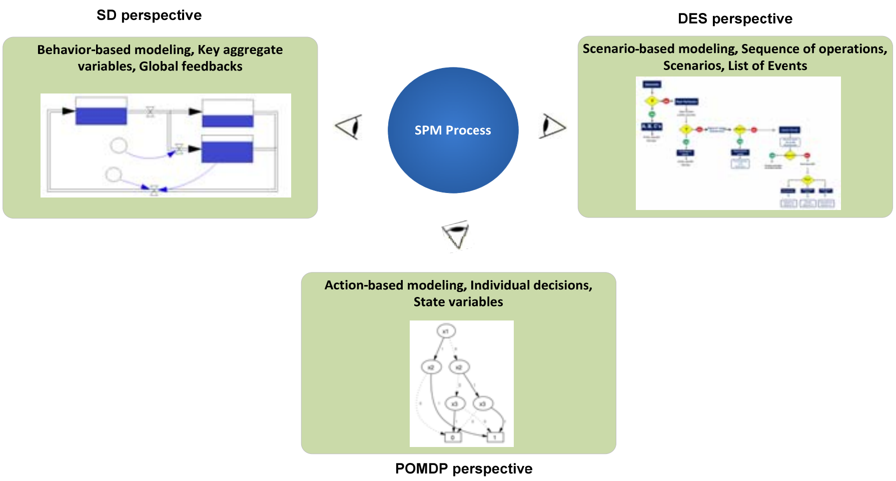


Figure 1. Process perspectives of each simulation method

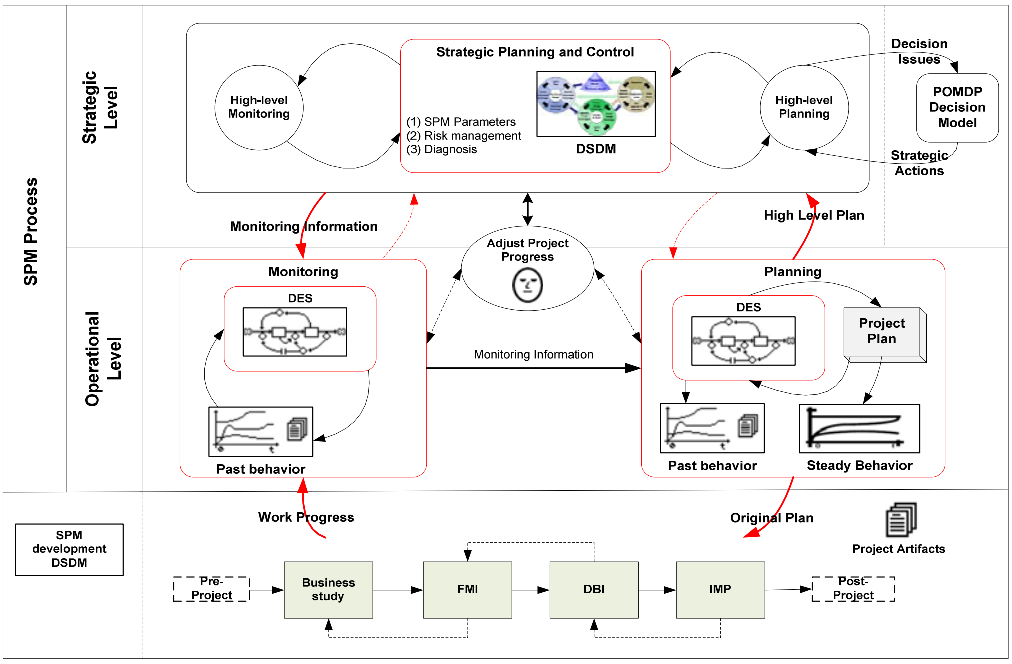


Figure 2. Multi-method simulation engine

2) Simulation engine

Simulation engine is formed by interrelated operability of DES and SD. The simulation engine is responsible to provide the basis for dynamism of simulation events. DES is adequately rich to develop simulation system, but on the other hand the lack of high-level abstract view of simulated environment makes it insufficient to bring in the critical characteristics of SPM. For this reason SD complements the operability of DS that allows the simulation system to be a strategic planning platform for SPM practice. Figure 2 illustrates the elements of multi-method simulation engine.

B. Strategic Decision Breakdown Framework for SPM

Strategic decision breakdown is an approach to model decision structure properly. In this approach the decision structure would be categorized by domain, objective and transformation. To address SPM decisions in the proposed framework with strategic perspective, type of decisions will be identified. These types are stereotypes of SPM activities according to SPM methodology specification. With identification of decision stenotypes, objectives and respectively the transformation function would be determined. Transformation is a mapping function that links a decision frame into related operational work breakdown structure. As illustrated in figure 3, the decision design process from strategic level to operational activities is depicted.

IV. Implementation

A. Generating Policy

For the proposed framework given the follow definition for POMDP model as S is the set of states, A is the set of actions and O is the set of observations:

S1= **phaseproceeding**, S2= **phasedone**

A1= **noact**, A2= **hire**, A3= **fire**, A4= **planreview**, A5=

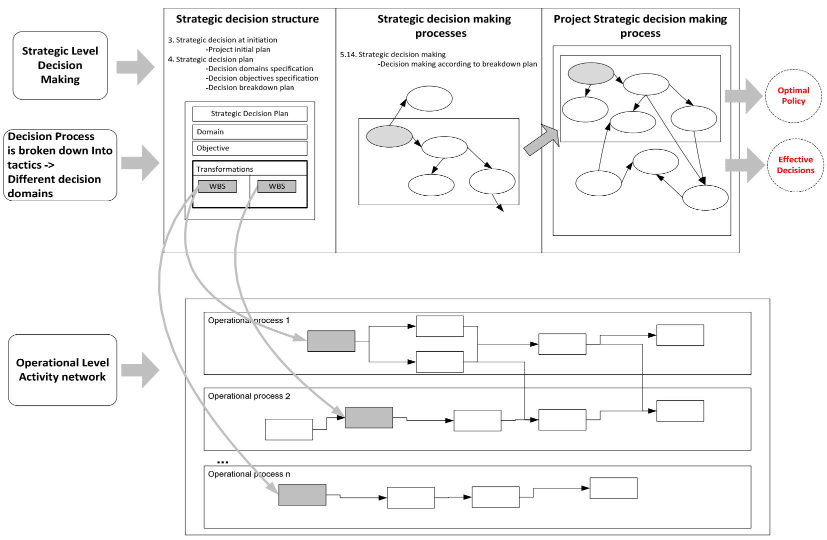


Figure 3. functional design of solution framework

buytool , A6= **determineiteration**

O1= **slowphaseprogress**, O2= **lowquality**,

O3= **behindschedule**, O4= **lowbudget**

The set of actions, observations and transition functions are elicited according to [15] risk prioritization, for the considerable actions and related observations. The action set labeled as "action stereotypes" helps to categorize the decision domains and to model the decision process effectively. Definition of transition functions are based on (1) and (2) respectively for actions and observations:

$$\mathrm{P(action)} = \frac{\mathrm{observ}+0.5\times\mathrm{semiobserv}}{\mathrm{total\ observations}} \quad (1)$$

$$\mathrm{P(observ)} = \frac{\mathrm{risks}+0.5\times\mathrm{semirisks}}{\mathrm{totalriks}} \quad (2)$$

P(action) is probability function of action over states, "observ" is the number of related observations, "semiobserv" is the number of semi-related observation, P(observ) is the probability function of observation over states and risks are related risks to the observations, "semiriks" is semi-related risks. Table II shows the project risk list.

We have a set of states, but we could never be certain where we are. A way to model this situation is to use probabilities distribution over the belief states. For better management of SPM process phases, the phase is divided into two states, "phaseproceeding" state which implies the process of the phase and "phasedone" which implies the phase is done. In a real SPM process each phase could be different dependant of manager's strategy but for formulating the process the same situation is considered for all phases of SPM. Therefore, the probability distribution over the two states is, Pr(s = phaseproceeding) = 0.50, Pr(s = phasedone) = 0.50 where s= state at time t. In this model there are advantages which would reduce the complexity of the algorithm of finding an optimal policy; 1-we know the initial belief point and 2-we know the initial action 3-belief state transition is one-way which only transition is from "phaseproceeding" to "phasedone". These three conditions of the model reduce the complexity of an exponential algorithm. There are 6 actions and 4 observations, according to (3):

$$\begin{aligned}&\text{Number of policies}\\&= (\text{number of actions})^{(\text{number of observations})}\\&= 6^4 = 1296\end{aligned} \quad (3)$$

It is a considerable large number to find an optimal policy from 1296 existent policies.

TABLE II.

PROJECT RISK LIST

Risk ID	Risk name	Nature
R01	Low budget	Cost and time
R02	Infractions against law	Contract
R03	Low communication and advertising for the show	User/ customer
R04	Unsuitable cast	Organization
R05	Unsuitable ticket price-setting	Strategy
R06	Unsuitable rehearsal management	Controlling
R07	Cancellation or delay of the first performance	Cost and time
R08	Poor reputation	User/customer
R09	Lack of production teams organization	Organization
R10	Low team communication	Organization
R11	Bad scenic, lightning and sound design	Technical performance
R12	Bad costume design	Technical performance
R13	Low complicity between cast members	Technical performance
R14	Too ambitious artistic demands compared to project means	Requirements
R15	Few spectators/lukewarm reception of the show	User/ customer
R16	Technical problems during a performance	Technical performance
R17	Low cast motivation	Organization
R18	Unsuitable for family audiences	Strategy
R19	Low creative team leadership	Controlling

Algorithm SARSOP

1. Initialize the Γ set of α-vectors, representing the lower bound $\underline{V}$ on the optimal value function V*. Initialize the upper bound $\overline{V}$ on V*.
2. Insert the initial belief point b_0 as the root of the tree T_R.
3. repeat
4. SAMPLE(T_R, Γ).
5. Choose a subset of nodes from T_R. For each chosen node b, BACKUP(T_R, Γ, b).
6. PRUNE(T_R, Γ).
7. until termination conditions are satisfied.
8. return Γ.

Figure 4. SARSOP algorithm

$$E^{\pi}(b_0) = \sum_{t=0}^{\infty} \gamma^t E[r(s_t, a_t)|b_0, \pi] \quad (4)$$

Where in (4), π is the policy, 0<γ<1 is discount factor, r is reward function, b0 is initial belief state and Eπ is expected value for policy π.

Then the optimal policy would be (5):

$$\pi^* = \underset{\pi}{\operatorname{argmax}} (E^{\pi}(b_0)) \quad (5)$$

Algorithm used to find the optimal policy is SARSOP [28] as described in figure 4.

1) Determining the optimal policy

The optimal policy for the simulation framework is described in table III. The transition of belief state with Piecewise linear and convex strategy, is converted into partitions, the belief space (state=phasedone).

Table III shows an optimal policy for this framework since there are only two states, belief state can be represented with a single value. In doing so it is not much more than a table lookup and using of Bayes Rule.

Generally finding an optimal policy over the POMDP is a very complex calculation form the complexity of algorithm chosen over an infinite number of horizons for the purpose. One of major issues in computing the optimal policy over belief states is the continuity. In finding POMDP optimal policy, it is more effective to divide the continuous belief space into several partitions and then to assign one action for each of the partitions. The set of partitions is resulted from the calculation of policy from infinite horizons and see the intersection for each of action-observation set of lines resulted from the value function called Piecewise linear and convex (PWLC). Figure 5 shows the visual PWLC presentation of computed optimal policy over the belief state partitions for the framework. The Y axis accounts for value of action and the x axis accounts for belief space probability distribution. Briefly, this figure illustrates the action segmentation for the considered belief space whereas in this case is the project progress.

2) Policy graph of POMDP

Policy graph is another form of a policy presentation for acting in a POMDP. A finite state controller, which each node of the graph is an associated action, and the edge out of the node going to other node is each observation that is possible. For this framework, a “policy graph” is shown in figure 6. Since the actual graph is very complicated, for simplification, maximum branches starting from a node to show is implemented. Figure 6 illustrates the same policy with 1 (a) and 2 (b) maximum branch(es) starting from a node. As it is perceived figure 6 (b) with only 2 branches out even has a complex structure for this framework and is presented to demonstrate the actual complication of the policy graph. This graph on the other hand provides a vision and clear visual for the analyzer to have a better insight on actions, observations and their impact on decision process. Also policy graph reveals the central tendency of decision; nevertheless this strategy makes the complexity of POMDP mitigated. The legends of symbols used in figure 6 are, A as action, O as observation and Y as state. For all symbols array number, the value in parentheses starts from zero and the value beside O is the probability of observation and Y is the belief point value.

TABLE III.
OPTIMAL POLICY OVER CONTINUOUS BELIEF

Partition No	Pr(state=phasedone)	Action
1	0.0000 to 0.3607	A2
2	0.3607 to 0.4537	A5
3	0.4941 to 0.6523	A3
4	0.6523 to 0.7566	A6
5	0.7566 to 0.7882	A1
6	0.5037 to 1.0000	A4

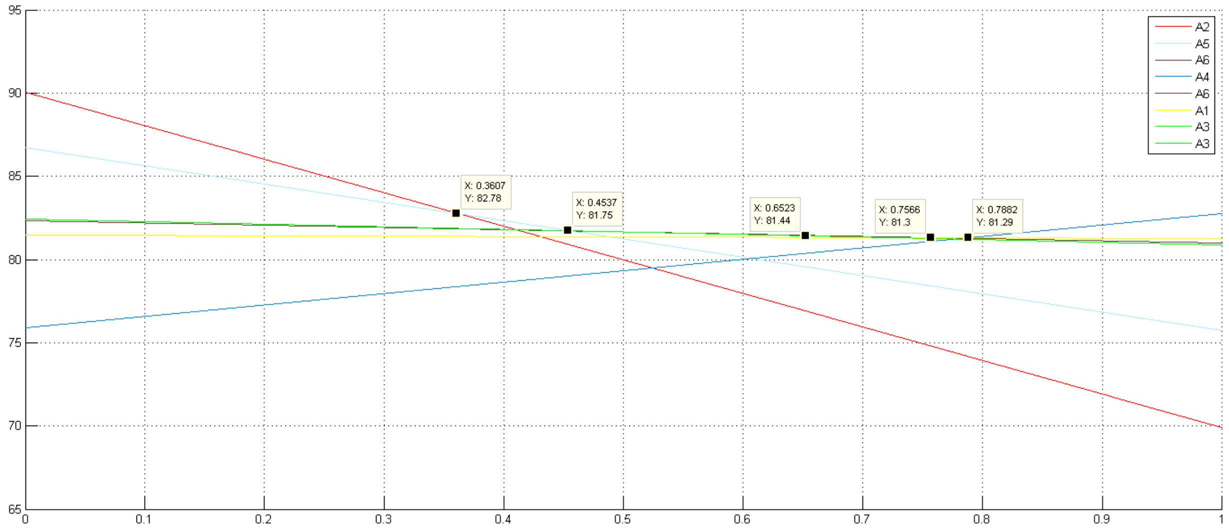
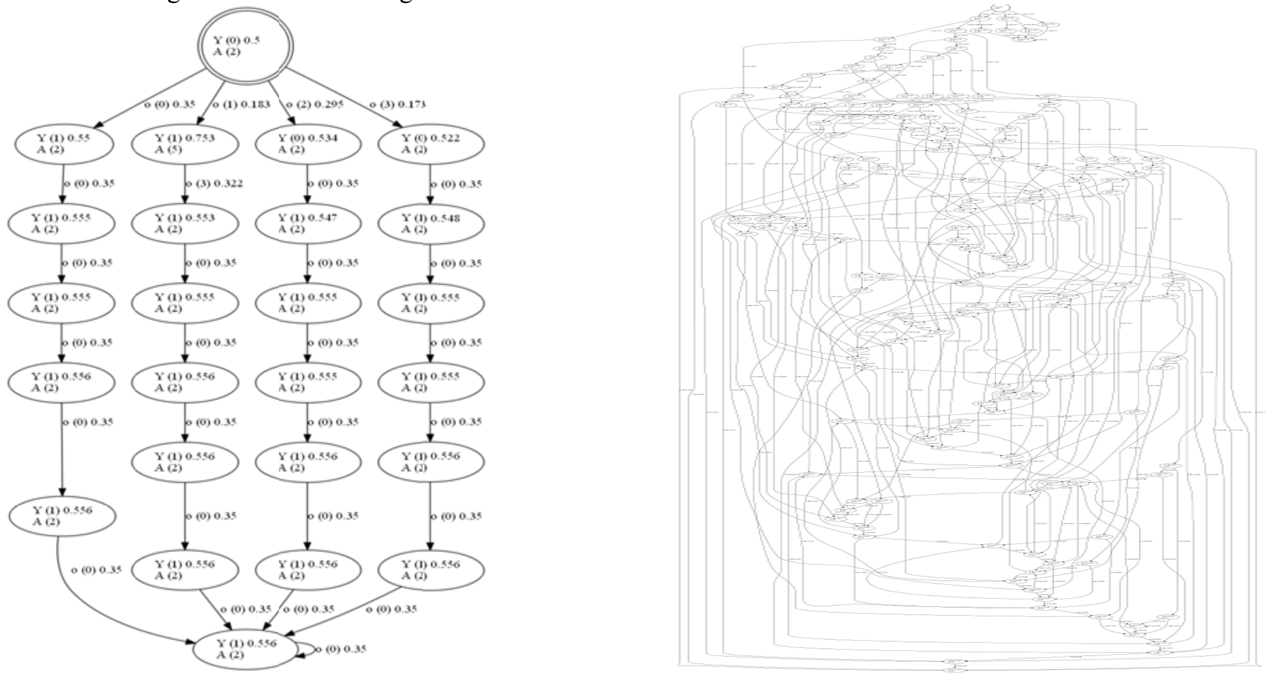


Figure 5. PWLC Visualization of optimal policy

V. Discussion

Policy based decision paradigm is part of a comprehensive decision support framework to open new horizons over SPM decision modeling. This feature is employed by the specified policy of POMDP to model the decision process and evaluate decision values. The framework helps practitioner to adjust their short-term perspectives over SPM process and see their actual decision feedback with regards to project constraints. The significance of decision modeling and an evaluation course for SPM decision process that roots from the complexity of this practice, inculcates that constructing a decision modeling framework is complicated. The proposed framework intends to form a different decision paradigm system from synthesis of decision support and decision management features. Figure 7 shows the implemented decision paradigm with the proposed framework. Decision management systems automate operational decisions (in other words they take actions); they mitigate the burden of decision making but restrict the freedom of users [29]. On the other hand decision support systems only provide recommendations for users and don't interfere in the process of decision making. By combination of these two systems, it is possible to develop a strategic level decision modeling framework by a sound foundation. Although as the optimal policy demonstrated the feasibility of such a framework, but the framework should not be taken as a mere decision management system. The existence and cooperation of SPM expert is necessary for ultimate assessment of the framework performance and course of action determination. SPM and related knowledge areas which are the root for decision modeling and new decision



(a) (b)

Figure 6. A policy graph with maximum of 1 branch (a) and 2 branches (b) out for optimal policy

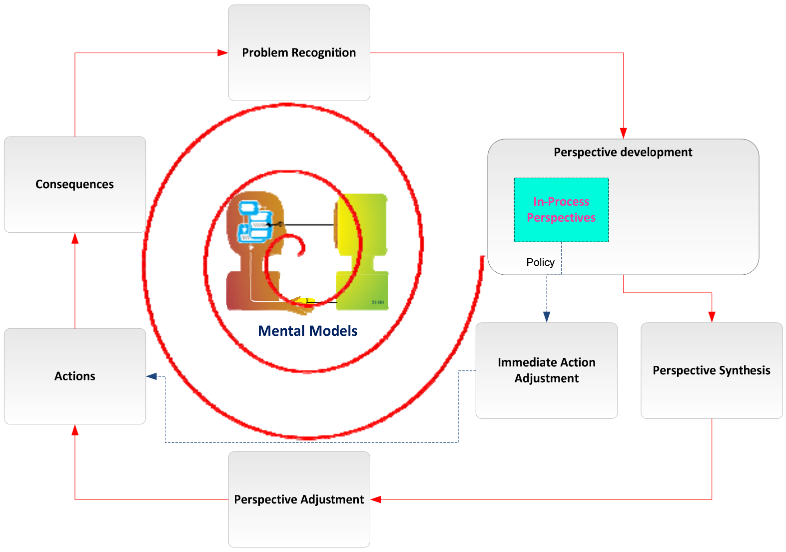


Figure 7. Policy based decision paradigm

paradigm embedment are illustrated in figure 8. In figure 8, the knowledge of SPM is categorized with planning and decision making areas for the purpose.

VI. Conclusion

The presented framework provides different views of SPM training, knowledge management, which were hardly considered in the existing approaches. These views are ranged from strategic, tactical, managerial and operational dimensions of SPM experiential knowledge.

The intention of implementing POMDP into the framework is to deal with complex aspect of SPM decision making process in which provides tactics and principles to evaluate decision values. SD with underlying basis of simulation supported by DES, provides a comprehensive simulation engine that on one hand makes the possibility of developing an operational framework upon the conceptual architecture and on the other hand transforms the simulation framework into a strategic planning-training platform. The framework brings on a delicate feature for SPM practitioner which is called in-process decision support. With this feature it is possible to assess the decision issues and deal with them according to the designated strategy in a real time fashion.

With integration of expert systems into the framework, the idea of reaching for having common features of decision support systems and decision management systems entirely will be accomplished. The goal is to develop a framework that involves in planning and decision making stages of SPM. The role of an expert system in the framework would be to specify structure for long-term perspective adjustment and help practitioners to acquire knowledge from past experiences. Also it would be possible to facilitate the tacit-explicit knowledge conversion of software project management.

Acknowledgement

Appreciation for direction and supervision of Dr

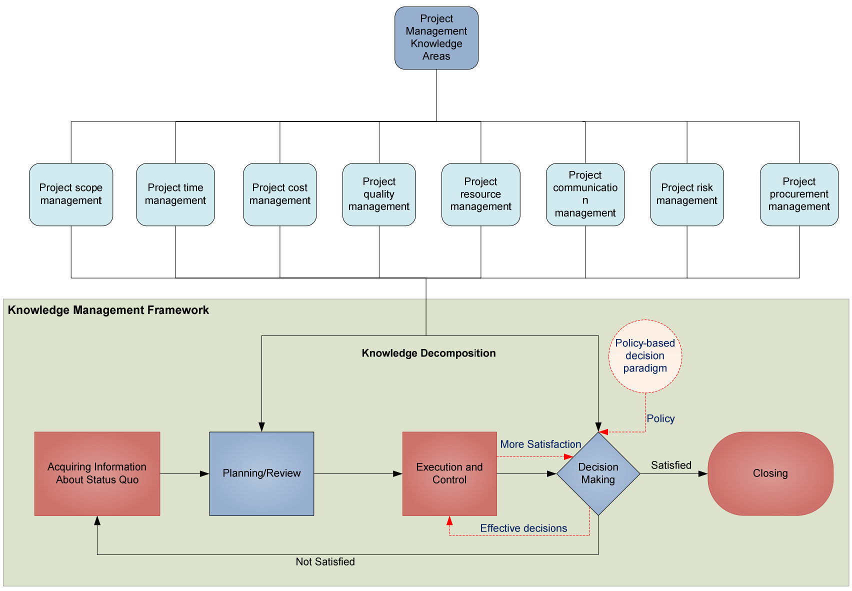


Figure 8. Knowledge management and new decision paradigm for SPM

Nazri Kama is necessary to acclaim. Also special thanks to Advanced Informatics School, UTM for cooperation and providing resources. The financial of this project is supported by Ministry of Higher Education Malaysia and Universiti Teknologi Malaysia under Vot No: 07J87.

REFERENCES

[1] J. Reason, *Human error*: Cambridge university press, 1990.

[2] D. Rodríguez, M. Satpathy, and D. Pfahl, "Effective Software Project Management Education through Simulation Models: An Externally Replicated Experiment," in *Proceedings of 5th International Conference on Product Focused Software Process Improvement (PROFES 2004)*, 2004, pp. 287-301.

[3] S. Cho "An exploratory project expert system for eliciting correlation coefficient and sequential updating of duration estimation," *Expert Systems with Applications, Elsevier,* vol. 30, pp. 553-560, 2006.

[4] A. Tizkar Sadabadi, "A Decision Support System for Game-Based Simulative Environment of Software Development Project Management," *International Journal of Machine Learning and Computing,* vol. 2, pp. 173-177, 2012.

[5] D. Power (2006, Dec 19th). *How should decision support projects be managed.* Available: http://dssresources.com/faq/index.php?action=artikel&cat=&id=166&artlang=en

[6] P. Donzelli, "A Decision Support System for Software Project Management," *IEEE SOFTWARE,* vol. 23, pp. 67-75, July/August 2006.

[7] H. S. Neap and T. Celik "A knowledge-based system for determination of marginal value of building projects," *Expert Systems with Applications,* vol. 21, pp. 119-129, 2001.

[8] A. P. Olteanu, "A Decision Support System (DSS) for Project Management in the Biodiesel Industry," *Informatica Economică,* vol. 15, pp. 189-202, 2011.

[9] G. H. Janczura and I. Golinska, "Decision support system for choosing model for a software development life cycle," *Operations Research and Decisions,* vol. 1, pp. 61-77, 2010.

[10] M. Plaza and O. Turetken, "A model-based DSS for integrating the impact of learning in project control," *Decision Support Systems,* vol. 47, pp. 488–499, 2009.

[11] S. Antony and R. Santhanam, "Could the use of a knowledge-based system lead to implicit learning?," *Decision Support Systems,* vol. 43, pp. 141–151, 2007.

[12] S. Besir and K. U. Birant, "A Case for Decision Support Systems on Project Management," in *ICT Innovations,Web Proceedings*, Ohrid, Macedonia, 2010, pp. 71-76.

[13] H. L. Yang and C. S. Wang, "Recommender system for software project planning one application of revised CBR algorithm," *Expert Systems with Applications,* vol. 36, pp. 8938–8945, 2009.

[14] A. Bollin, E. Hochmüller, R. Mittermeir, and L. Samuelis, "Experiences with Integrating Simulation into a Software Engineering Curriculum," in *25th IEEE Conference on Software Engineering Education and Training*, 2012, pp. 62-71.

[15] C. Fang and F. Marle "A simulation-based risk network model for decision support in project risk management," *Decision Support Systems,* vol. 52, pp. 635-644, 2012.

[16] S. T. Huang, W. H. Lin, and M. C. Hsu, "Embracing Business Context in Pedagogical Simulation Games-A Case with Process Disciplined Project Management," in *CSEETW '08 Proceedings of the 2008 21st IEEE-CS Conference on Software Engineering Education and Training Workshop*, 2008, pp. 9-12.

[17] E. O. Navarro, "SimSE: A Software Engineering Simulation Environment for Software Process Education," Doctoral, Donald Bren School of Information and Computer Sciences, University of California, Irvine, 2006.

[18] R. T. Mittermeir, E. Hochmüller, A. Bollin, S. Jäger, and M. Nusser, "AMEISE - A Media Education Initiative for Software Engineering: Concepts, the Environment, and initial Experiences " in *Proceedings International Workshop ICL - Interactive Computer Aided Learning*, Villach, 2003.

[19] A. Bollin, E. Hochmuller, and R. T. Mittermeir, "Teaching Software Project Management using Simulations," in *24th IEEE-CS Conference on Software Engineering Education and Training (CSEE&T)*, 2011, pp. 81-90.

[20] Z. Gao and C. Xie, "The Study of Content Simulation Using in the Software Project Management Teaching," in *2010 Second International Workshop on Education Technology and Computer Science*, 2010, pp. 576 - 578.

[21] R. Salas, J. L. Wildman, and R. F. Piccolo, "Using Simulation-Based Training to Enhance Management Education," *Academy of Management Learning & Education,* vol. 8, pp. 559–573, 2009.

[22] D. Nembhard, K. YIP, and A. Shtub, "Comparing Competitive and Cooperative Strategies for Learning Project Management," *Journal of Engineering Education,* pp. 181-192, 2009.

[23] D. Rodríguez, M. A. Sicilia, J. J. C. Gallego, and D. Pfahl, "e-Learning in Project Management Using Simulation Models: A Case Study Based on the Replication of an Experiment," *IEEE TRANSACTIONS ON EDUCATION,* vol. 49, pp. 451-463, NOVEMBER 2006 2006.

[24] A. Targowski, "From Data to Wisdom in the Global and Civilizational Context the Cognitive Perspective," in *ENTERprise Information Systems*, J. E. Q. Varajão*, et al.*, Eds., ed Viana do Castelo, Portugal: Springer, 2010, pp. 21-30.

[25] B. Zeigler, T. G. Tag Gon Kim, and H. Praehofer, *Theory of Modeling and Simulation,Second Edition:Integrating Discrete Event and Continuous Complex Dynamic Systems*, second ed.: Academic Press, 2000.

[26] J. D. Sterman, "System Dynamics Modeling," *California management review,* vol. 43, pp. 8-25, 2001.

[27] L. P. Kaelbling, M. L. Littman, and A. R. Cassandra, "Planning and acting in partially observable stochastic domains," *Artificial intelligence,* vol. 101, pp. 99-134, 1998.

[28] H. Kurniawati, D. Hsu, and W. S. Lee, "SARSOP: Efficient Point-Based POMDP Planning by Approximating Optimally Reachable Belief Spaces," in *Robotics: Science and Systems*, Zurich, Switzerland, 2008, pp. 65-72.

[29] J. Taylor and N. Raden, *Smart Enough Systems: How to Deliver Competitive Advantage by Automating Hidden Decisions*. Upper Saddle River, NJ, USA: Pearson Education, 2007.

Ali Tizkar Sadabadi obtained his bachelor degree at Islamic Azad University of Tabriz (IAUT) in Software Engineering in 2004, master degree at State Engineering University of Armenia (SUEA) in Informatics and Computer Systems in 2008 and currently is a PhD candidate at Universiti Teknologi Malaysia (UTM) in the field of Informatics. He lectured courses and conducted several projects in the field of computer science and information technology. His major research areas are software development and the implementation of simulation and expert systems in applicative fields.

Nazri Kama obtained his first degree at Universiti Teknologi Malaysia (UTM) in Management Information System in 2000, second degree in Real Time Software Engineering at the same university in 2002 and his PhD at The University of Western Australia (UWA) in Software Engineering in 2010. He has a considerable experience in a wide range on Software Engineering area. His major involvement is in software development. Currently, he is holding a position of Chief Executive Officer of a Spin-off company under his University.