Multi-method simulation approach for software project management training in-process decision support system: a conceptual-operative framework

Ali Tizkar Sadabadi, Nazri Kama, Faizul Azli Abd Ridzab
Department of Software Engineering, Advanced Informatics School, Universiti Teknologi Malaysia,
Kuala Lumpur, 54100, Malaysia
tsali2@live.utm.my, nazrikama@ic.utm.my

Abstract—Software project management (SPM) is a discipline that comprises of different topics, practice and theories. There are two dimensions in the knowledge of SPM, theoretical or concepts of SPM and; practical or experience of SPM. These two dimensions although grow separately but come across at one point that is experiential knowledge of SPM. To present these dimensions through a proper training, a practitioner needs to have a proper view on SPM process. In this paper we present a new framework for practicing SPM through a distinct simulation technique to bring on an in-process decision support system, supporting the proper training approach. The framework developed based on three different methods of simulation technique, discrete event simulation (DES), system dynamics (SD) and partially observable Markov decision process (POMDP).[1]

Keywords—Simulation model, software project management, computer based training (CBT), multi-method framework.

I. Introduction

This paper would present a training framework for SPM practitioner that improves his/her knowledge of SPM, in particular experiential knowledge. This improvement occurs in a real-time fashion which in this paper called in-process decision support. The motivation for developing such a framework, comes from the facts have been introduced and mentioned as a challenge in literature of software engineering training [1-7], SPM training needs different approaches than traditional, inflexible and unchallenging approaches have been introduced. The nature of software projects which comprises intensive complexity, intangible products that interconnected with business dynamics and complexity, management capability of these factors needs special capacity and experience. On the other hand the mindset of practitioners of this field that majorly sheer considered for management and less attention to the technology is another fact that has an impact on SPM proper training.

Simulation accompanied with game techniques is introduced to define a different approach for implementing IT technologies and to overcome that shortcoming as mentioned for SPM training plights. In this paper, the exploitation of simulation technique is in a modality which has not been implemented in current works, which mainly focused on flat scenarios, and covers the major issues of SPM. Also, the framework provides different views on SPM that each of these views provides distinct level of knowledge. We defined 4 views over SPM which are: explicit-direct, operational, tactical and strategic. Explicit-direct view only deals with explicit type of knowledge and other views mainly deal with tacit type of knowledge. Briefly, the framework prepares a basis for in-process decision support system. This feature of the framework would be used as an evaluation strategy for assessment of the framework user's decision values.

II. Simulation in Software Engineering Training

Current Approaches that have been developed by many scholars [1-3, 5-9] represented an

[1] The financial of this project is supported by Ministry of Higher Education Malaysia and Universiti Teknologi Malaysia under Vot No: 00K01.

implementation of simulation framework/model in the process of software engineering education as well as SPM training. In general simulation framework/model is developed based on traditional approaches which construct concepts of SPM. SimSe, SESAM, OSS as three of major simulation frameworks have been selected for this paper to present the contribution they made in the respective field. There are recent works in this field with implementation of simulation technique as educational approach but mainly they developed their work based on the facts which are presented in these frameworks. Table I shows the comparison of the three selected major frameworks based on simulation models. It is considered from the study on the current works that their presented framework/model on the level of views, only operational view is provided.

III. Simulation techniques implemented in different contexts

A. *Simulation implementation from different perspectives*

There are three major simulation techniques which they cover three distinct aspects of simulation process: 1- discrete event simulation 2- system dynamics 3- partially observable Markov decision process (POMDP).

DES is responsible for providing a low level and operational part of simulation. Components of this method are as follow: clock, event list, random number generator. Logic for this method is defined by a main loop with ending condition(s). The simulation engine and logic is only sufficient to describe basic function of simulation.

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 [10]. 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 [11]. An exact solution to a POMDP yields the optimal action for each possible belief over

TABLE I. COMPARISON OF FRAMEWORKS BASED ON SIMULATION MODEL FOR SPM TRAINING

Framework	Discipline Coverage	Strength	Limitation
SimSE	Software development process	• Supports detailed software engineering education • Support project management training • Support artifact analysis. • user-friendly interface (interactive & attractive)	• No decision support feature • The training level is low (little to gain for software engineering skills) • No tactical view provided • No strategic view provided
SESAM	software engineering education	• Support various aspects of software engineering • Support project management training • Natural language user interface	• Communication with interface requires high skill of software engineering • No decision support feature • The interface is text-based (not user-friendly) • No tactical view provided • No strategic view provided
OSS	software engineering education	• Multimedia based training • Case study based training • Simple interface	• No decision support feature • project management training is minor • No tactical view provided • No strategic view provided

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 is accountable to bring on specific level of view over simulation process. Each of these methods operates at different level of abstraction and comprises of distinct elements. The simulation methods and their correspondence to level of abstraction and operation are illustrated in fig. 1.

Respectively, DES is the machine level code which is the basis for constructing the simulation system. SD which entails the highest level of simulation perspective that provides an executive managerial view and level of understanding from the system behavior. Yet the multi-method simulation approach is not coherent and there is a considerable big gap between these two levels of simulation. Then POMDP is implemented to fill out this gap and to bring about a 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. Tactical view is indented to provide for senior managerial and decision-based level of understanding over SPM process.

IV. A SIMULATION FRAMEWORK ARCHITECTURE FOR SPM TRAINING

A. SPM methodology for basis of modeling

it was decided to choose Dynamic systems development method (DSDM) method [12] to define the simulation process basis. DSDM does specify concrete results for each task and for each one of the three phase groups (FMI, DBI and Implementation).

Therefore we divided major stages of project development into four not including pre-project and post-project stages. The definitions of phases which will be introduced afterwards are as follow: Phase 1- feasibility (either could be defined here or in Initial project definition stage) and business study (FSB). Phase 2- Functional Model Iteration (FMI). Phase 3- Design & Build Iteration (DBI). Phase 4- Implementation (IMP).

B. Multi-method simulation framework for SPM training

Based on SPM methodology and multi method simulation technique, a SPM training simulation framework is developed. This framework is accountable to represent a simulation environment for SPM practitioners. The framework brings on the critical characteristics of software projects. These characters are defined from different views which are imaginable over SPM training environment.

The framework implements, three simulation methods as mentioned before DES, SD and POMDP. Integration of these methods for the framework is as follow:

1) Simulation engine

Simulation engine is formed by multi-method simulation model, DES and SD. The simulation engine is responsible to provide the basis for realization of simulation environment. DES provides the machine level simulation language to design. DES is adequately rich to develop simulation system, and then the necessity to implement this technique is obvious. But on the other hand the lack of high-level strategic view of simulated environment makes it insufficient to bring in the critical characters of SPM. For the best modeling we embedded SD technique. This allows evaluating the simulation system for senior SPM practitioners. Fig. 2 illustrates the elements of multi-method simulation engine has been shown.

2) Simulation logic

Simulation logic is the communication language of simulation elements which are simulation states. The language should be able to define the casual-effect chain between events which have impacts on states. Prediction of causes and which effect they will have impact on and to chase this series of chains of events is somehow needs an abstraction of events level and gain a judicious and conceptual view of states. The technique which is implemented for this purpose is POMDP.

Simulation logic with POMDP provides a pattern by an optimal policy to determine an effective approach of SPM process. This process is majorly based on decisions of manager, thus there is a requirement to define a model to evaluate these decisions. The evaluation is done based on in-process decision support system feature that this model will be implemented to provide a definite pattern based on stochastic process to evaluate the decision process and decision values.

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:

States are S1= phaseproceeding and S2=phasedone

Then actions are:

A1=noact, A2=hire, A3=fire, A4=buytool, A5= planreview, A6= determineiteration

And observations are:

O1= slowphaseprogress, O2= lowquality, O3= behindschedule, O4= lowbudget

We elicited the set of actions, observations and transition functions according to [13] risk prioritization, defined the proper actions and observations. Definition of transition functions are based on (1) and (2) respectively for actions and observations:

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

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

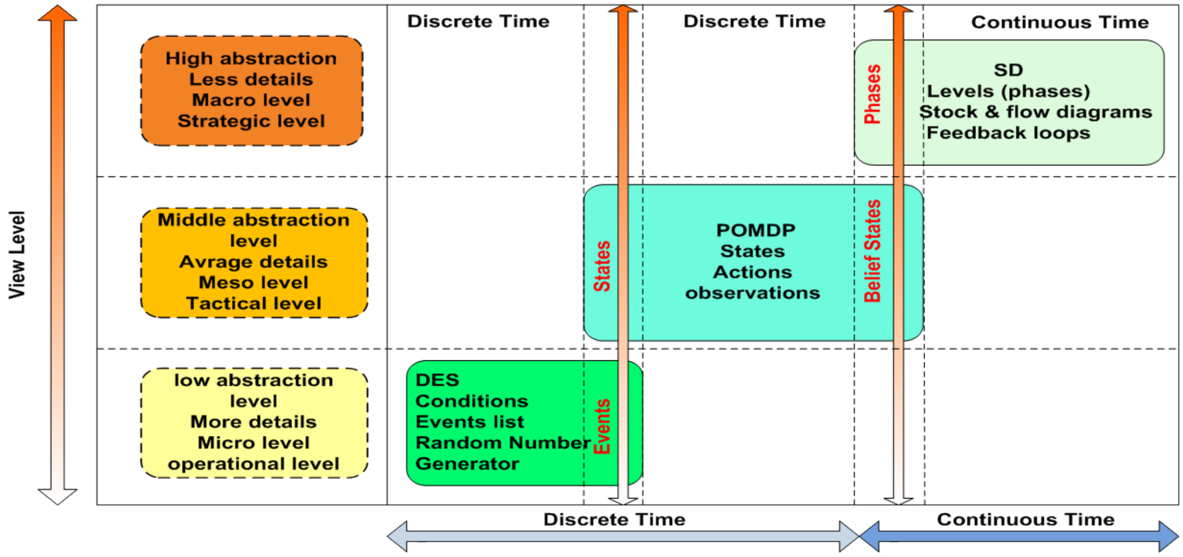

Fig. 1. Simulation methods and their correspondence to operational and abstract level.

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.

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
R20	Low creative team reactivity	Controlling

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, we divided the phase 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 we considered the same situation for all phases of SPM. Therefore here as follow is the probability distribution over the two states. Pr(s = phaseproceeding) = 0.80, Pr(s = phasedone) = 0.20 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 meaning only the transition is from "phaseproceeding" to "phasedone". These three conditions of the model reduce considerably the complexity of algorithm which is exponential. There are 6 actions and 4 observations,

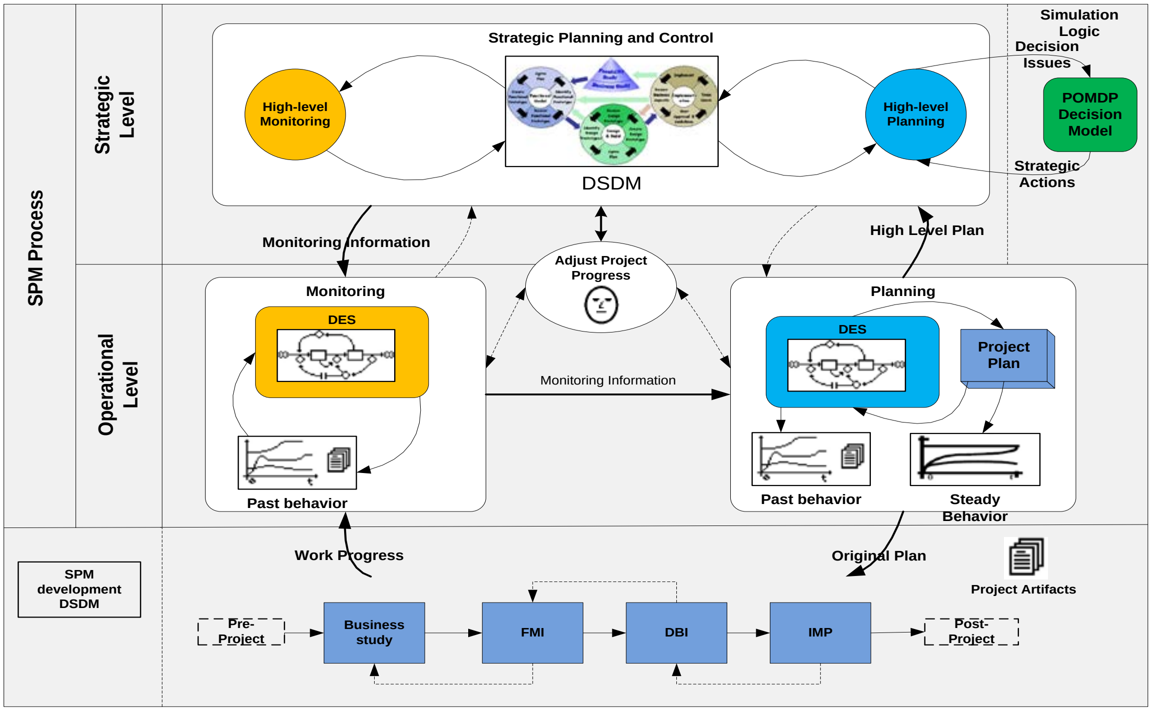

Fig. 2. The multi-method simulation engine.

according to (3):

Number of policies=(number of actions)^(number of observations)=6^4=1296 (3)

It is considerably big number to find an optimal policy from 1296 existent policies.

(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):

$$\pi^{*} = \underset{\pi}{\mathbf{argmax}} \llbracket(\mathrm{E}\rrbracket^{\pi}(\mathrm{b}_{0})) \tag{5}$$

For a simple description of the algorithm to find the optimal policy, it comprises of stages which are explained as follow: 1- Use the policy to select action for current belief state. 2- Execute the action. 3- Receive an observation and immediate reward. 4- Update the belief state using current belief, action and observation. 5- Repeat.

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) as represented in table III.

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. Fig. 3 shows the visual Piecewise linear and convex (PLWC) presentation of computed optimal policy over the belief state partitions for the framework.

TABLE III. THE OPTIMAL POLICY OVER CONTINUOUS BELIEF STATE

Partition No	Pr(belief space)	Action
1	0.000 to 0.100	A1
2	0.100 to 0.200	A2
3	0.200 to 0.300	A4
4	0.300 to 0.400	A6
5	0.400 to 0.500	A2
6	0.500 to 0.600	A5
7	0.600 to 0.700	A6
8	0.700 to 0.800	A3
9	0.800 to 0.900	A5
10	0.900 to 0.950	A6
11	0.950 to 1.000	A5

.

a) POMDP Policy

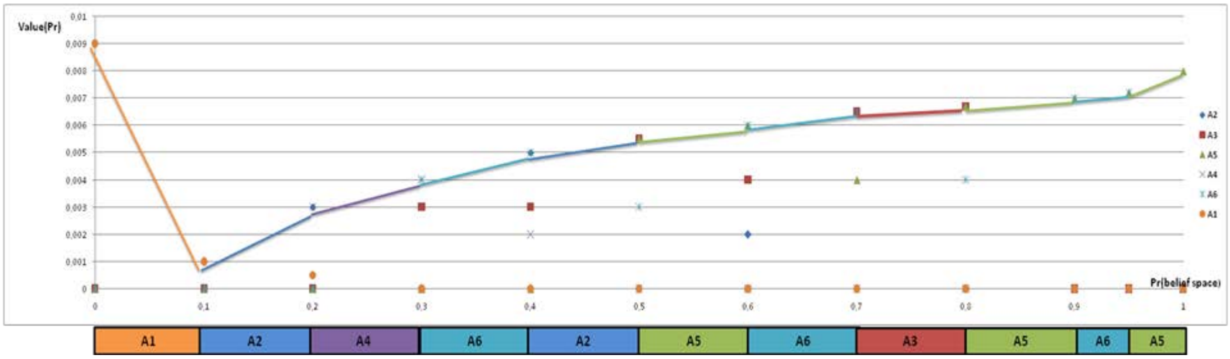

Fig. 3. The visualization of Piecewise linear and convex (PLWC) presentation of the optimal policy.

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 issue in computing the optimal policy over belief states is that it is continuous. 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 are 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 (PLWC).

b) Policy graph of POMDP

Policy graph is another form of a presentation policy 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 fig. 4. This graph on the other hand provides envisage and clear visual for the analyzer to have a better insight on the action, observations and their impact on decision process. Also this graph reveals the central tendency of decision in a visual fashion. This graph could be considered as a finite state machine; nevertheless this strategy makes the complexity of POMDP mitigated.

C. Operational simulation environment architecture

Simulation environment, based on described models, has 4 elements: (1); Simulation engine (2) Simulation logic;(3) Simulation states;(4) Simulation GUI. The operational, DES-based, simulation model architecture is depicted in fig. 5.

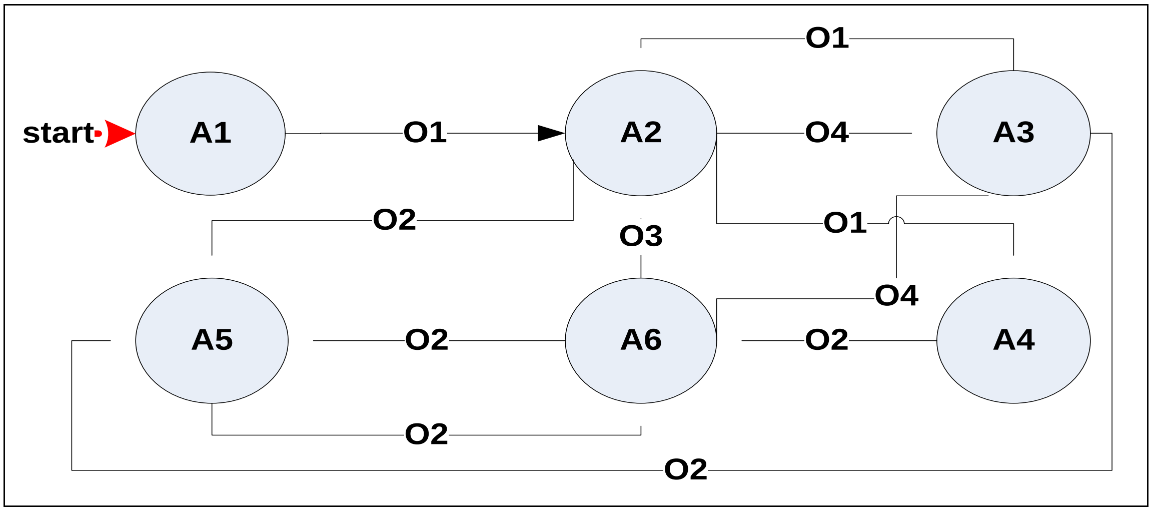

Fig. 4. The policy graph for optimal POMDP policy.

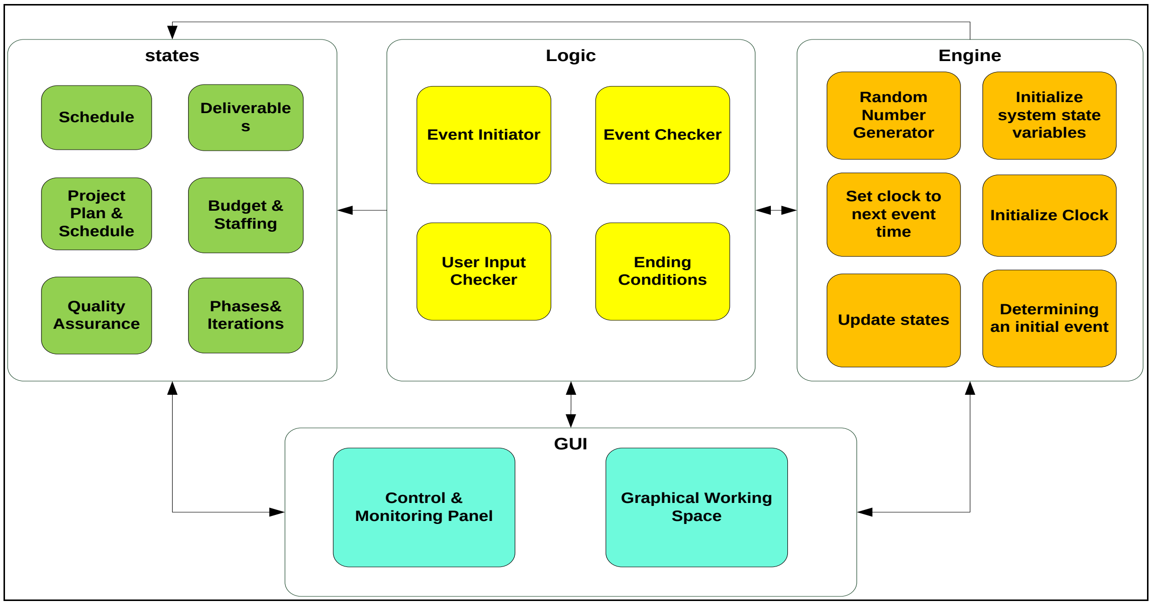

Fig. 5. The operational simulation environment architecture.

V. A CONCEPTUAL-OPERATIVE SIMULATION FRAMEWORK FOR SPM TRAINING

Based on the formulation and descriptions, now it is the time to combine the developed concepts and models into a functional framework. The integration process would result a novel framework which covers complex aspects of SPM training and knowledge acquisition. Fig. 6 describes the functional system of conceptual-operative simulation framework for SPM training. As illustrated in fig. 6 the framework is completely interactive and captures the decision made by the user over the runtime of the simulation process. The outcome of decisions would be stored in a variable to evaluate the value of the decision based on comparison with decision model which bring on a pattern for optimal decision policy. This variable would be the parameter to determine whether the process concluded with success or with failure.

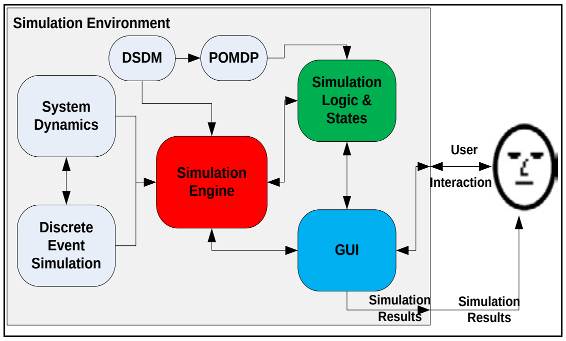

Fig. 6. Conceptual-operative simulation framework.

VI. CONCLUSION

The presented conceptual-operative framework provides different views of SPM training which were hardly considered in the existing approaches. These views are ranged from strategic, tactical 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 that provides tactical view and the basis to evaluate the decisions values. SD with underlying basis of simulation provided by DES, provides a comprehensive simulation engine which on one hand makes the possibility of developing an operational framework based on 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.

VII. FUTURE WORK

The framework is to be transformed into an operational solution. This solution would be based on interactive features of simulation environment complied with SPM training specifications. This framework provides a rich basis for in-process decision making support. That implies the distinct functions to capture online decision issues and deal with them in real-time manner. This capability is a measure to evaluate that the process is a success or a failure.

Another future work is appending an expert system model into this framework. This framework only provides in-process decision support capability. But for analyzing the results of simulation it is necessary to develop a system, by implementation of expert systems. The purpose is to provide off-process and analogy based analysis of simulation past results accumulated over time. Aside the off-process decision support, this approach will allow to append another view on the framework. This view is knowledgeable and provides for middle managerial and entails knowledge-based understanding level of the process. This approach will bring on an exclusive feature to the framework to operate on knowledge development dimension of knowledge management process.

ACKNOWLEDGEMENT

Appreciation for direction and supervision of Dr Nazri Kama is necessary to acclaim. Also special thanks to Advanced Informatics School, UTM for cooperation and providing resources.

REFERENCES

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

[2] 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.

[3] R. Salas, *et al.*, "Using Simulation-Based Training to Enhance Management Education," *Academy of Management Learning & Education*, vol. 8, pp. 559–573, 2009.

[4] S. T. Huang, *et al.*, "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.

[5] L. Davidovitch, *et al.*, "Simulation-based Learning in Engineering Education: Performance and Transfer in Learning Project Management," *Journal of Engineering Education*, 2006.

[6] E. O. Navarro and A. V. D. Hoek, "Design and Evaluation of an Educational Software Process Simulation Environment and Associated Model," in *In Proceedings of the Eighteenth Conference on Software Engineering Education and Training*, Ottawa, Canada, 2005a.

[7] E. O. Navarro and A. V. D. Hoek, "Software Process Modeling for an Educational Software Engineering Simulation Game," *Software Process Improvement and Practice*, vol. 10, pp. 311–325, 2005b.

[8] Lucko.G, *et al.*, "Harnessing the power of simulation in the project management / decision support aspects of the construction industry," in *Proceedings of the 2008 Winter Simulation Conference*, 2008.

[9] A. Bollin, *et al.*, "Teaching Software Project Management using Simulations – The AMEISE Environment: from Concepts to Class Room Experience," in *25th IEEE Conference on Software Engineering Education and Training*, 2012, pp. 85-86.

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

[11] L. P. Kaelbling, *et al.*, "Planning and acting in partially observable stochastic domains," *Artificial intelligence*, vol. 101, pp. 99-134, 1998.

[12] B. J. J. Voigt, "Dynamic System Development Method," Department of Information Technology, University of Zurich, Zurich20 January 2004.

[13] 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.

Creative Commons Attribution License 4.0 (Attribution 4.0 International, CC BY 4.0)