A Conceptual-operative Framework for in-process Decision Support of Software Project Management Practice

Ali Tizkar Sadabadi1, * and Nazri Kama

Department of Software Engineering, Advanced Informatics School, Universiti Teknologi Malaysia, Kuala Lumpur, 54100, Malaysia

tsali2@live.utm.my, nazrikama@ic.utm.my

*Corresponding Author

Abstract

The intention of this paper is to present an in-process decision support feature for Software project management (SPM). The significance of decision support in knowledge management of any discipline is undeniable and more prominent when it includes managerial capabilities. The specified in-process decision support in this paper astutely would address the distinct class of decision issues and accordingly tries to resolve them with an optimal policy. This feature would be accomplished with a multi-method simulation approach comprises of three different methods of simulation technique. The simulation methods implemented for the approach are discrete event simulation (DES), system dynamics (SD) and partially observable Markov decision process (POMDP). In this paper the composition of this framework and consequently the simulation results of POMDP policy which is operation-center for decision support would be presented.

Keywords: multi-method Simulation, software project management, decision support systems (DSS), in-process decision support, computer based training (CBT)

1. Introduction

SPM (Software project management) is basically defined on the process of decision making. It is the responsibility of managers to design this process and optimize it to obtain minimum cost and maximum production. The decision makings are based on resources and constraints which are planned for the target project and they could change hardly according to new requirements during the project progress. But what could be done accordingly to confront to the changes which are contingent in every project and basically to define an optimum plan based on optimum decisions which are based on deep understanding over the process, enterprise and recourses.

Decision making is an ability based on experience and knowledge that enables the leader of a process to success. This is a key point to find out a best practice to define for development of decision making capability. Since the nature of software projects is in a unique state which development process is complex and its intangibility makes the practitioners and managers unable to define solid models over the process.

It is evident that a novel and effective approach for modeling of the decision making process and evaluating the defined model over SPM is necessary. This model should be able to deal with high level of complexity and high change rate within software projects.
In this paper the aim is to present a framework which constructs the basis for modeling of decision making process over SPM. This framework incorporates knowledge management and knowledge engineering disciplines as well the methodology is supported by SPM theories. In the framework new terms for decision making are defined and the possibility of distinct levels of view over SPM would be realized.


DSS (Decision support systems) are mainly implemented as expert systems into SPM process. With study in the respective literature, scholars [1-8] stated the improvement of decision making by implementation of expert systems in SPM. As stated by Antony and Santhanam [9] with supporting decision-making ability, the use of knowledge-base system could implicitly improve the learning process. The use of expert systems is majorly to improve the decision making ability but consequently this could impact on the understanding over the process. Olteanu [10] addresses the necessity of implementation of DSS in biodiesel project management to identify all opportunities for improvement of decision value and lowering the production cost. (Janczura and Golinska [11] 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 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.” They applied cased –based reasoning technique for describing the project requirement accurately.

The application of DSS as a mean for leveraging the education scheme for SPM seems out of attention in the respective literature. The alienating factor for scholars in the SPM education or even any branch of science with connection with education to implement expert systems is based on the impact which expert system would have on education process. The question has been raised in the literature and responded accordingly. One answer claims the knowledge-based systems are able to improve the level of education by effecting the learning process implicitly (DSS) [9].

Practitioners need an opportunity to share, either with communication or retaining their experiences to develop the accumulated “embrained knowledge”. Hence it is necessary to consider such a system for converting the experience into expressions and retaining them in the knowledge-base.

Table 1 shows the characteristics of existing works with implementation of simulation and present a comparative style for their supported features. The table is to bring on the possibility of pointing out the features of existing works according to the research objectives and criteria. The initials of terms which are specified in the table are as follow:
E-D: explicit-direct view considered, O: operational view considered, M: managerial viewed considered, T: tactical view considered, S: strategic view considered, DS: decision support feature, In-P: in-process decision support considered, Off-P: off-process decision support considered, Y: yes, N: No, P: partly.

2.1. Summary of the Literature

With the stated facts, the definition of DSS should be done to address the decision issues in SPM precisely. The necessity for implementation of expert systems is undeniable. But what the literature has provided as the basis for justification and the benefits from embedding expert systems into SPM decision making process, is only the need for the knowledge-based systems which is considered as ordinary support for decision process. What the process needs and what the project managers expect are quite broader that what is presented in the existing works.

The intention is to define a comprehensive definition for a DSS to realize certain aspects on SPM process. The decision support in the literature of decision support systems is generally specified by Gachet [14] Using the relationship with the user as the criterion, differentiates passive, active, and cooperative DSS. In this paper decision issues with the concern of time would be divided into short-term and long-term. This paper tries to target the short term decision issues and their feedbacks that impact directly on actions. These issues are specified to be covered by in-process decision support feature. The important fact about these decision types is that their impacts are immediate and there is a volatile and critical opportunity to learn about the consequences of made decisions.

3. Simulation Techniques Implemented in Different Contexts

3.1. Simulation Implementation from Different Perspectives

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

DES is accountable 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 [15]. 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 [16]. 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.

3.1.1. 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 Figure 1. Respectively, DES is the machine level code which is the basis for constructing the simulation system. SD 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 gap between these two levels of simulation. POMDP is implemented to fill out this gap and to realize 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 managers.

<table>
<thead>
<tr>
<th>Name</th>
<th>Reference</th>
<th>Discipline Coverage</th>
<th>Explicit</th>
<th>Tacit</th>
<th>DS</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td>E-D</td>
<td>O</td>
<td>M</td>
</tr>
<tr>
<td>SimSE</td>
<td>Navarro, 2006</td>
<td>Software Development Process</td>
<td>Y</td>
<td>Y</td>
<td>N</td>
</tr>
<tr>
<td>SESAM</td>
<td>Drappa and Ludewig, 2000</td>
<td>Software Engineering Education</td>
<td>Y</td>
<td>Y</td>
<td>N</td>
</tr>
<tr>
<td>OSS</td>
<td>Sharp and Hall, 2000</td>
<td>Software Engineering Education</td>
<td>Y</td>
<td>P</td>
<td>N</td>
</tr>
<tr>
<td>AMEISE</td>
<td>Bollin et al., 2011</td>
<td>SPM Education</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
</tr>
<tr>
<td>PMT</td>
<td>Davidovitch et al., 2006</td>
<td>SPM Education</td>
<td>Y</td>
<td>Y</td>
<td>N</td>
</tr>
</tbody>
</table>
4. A Simulation Framework Architecture for SPM Training

4.1. SPM Methodology for Basis of Modeling

Dynamic systems development method (DSDM) [17] has been chosen as a methodology 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 and business study (FBS). Phase 2- Functional Model Iteration (FMI). Phase 3- Design & Build Iteration (DBI). Phase 4- Implementation (IMP).

4.2. Multi-method Simulation Framework for SPM Training

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

The framework implements, three simulation methods DES, SD and POMDP. The integration of these methods is as follow:

4.2.1. Simulation Engine: Simulation engine is formed by multi-method simulation model, DES and SD. The simulation engine manages the simulation states. DES is the fundamental for design and development of simulation process. DES is adequately rich to develop
simulation system, but on the other hand the lack of abstract view of simulated process makes it insufficient to design the critical characteristics of SPM. For the best modeling, SD technique is embedded. Figure 2 illustrates the elements of multi-method simulation engine.

4.2.2. 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 events chain is somehow needs an abstraction of events level and have a conceptual view over states. The technique which is accountable for this purpose is POMDP.

Simulation logic with POMDP provides a pattern by an optimal policy to determine an effective approach for SPM process. SPM is majorly based on decisions of manager, thus there is a requirement to define a model to evaluate these decisions. The evaluation is conducted through in-process decision support feature. This feature provides a pattern based on stochastic process to evaluate the decision process and compute decision values.

For the proposed framework given the following definition for POMDP model proposed 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

![Figure 2. The Multi-method Simulation Engine](image-url)
The set of actions, observations and transition functions are elicited according to [18] risk prioritization, defined the proper actions and observations. Definition of transition functions are based on (1) and (2) respectively for actions and observations:

\[
P(\text{action}) = \frac{\text{observ} + 0.5 \times \text{semiobserv}}{\text{total observations} + 0.5 \times \text{semirisks}}
\]

\[
P(\text{observ}) = \frac{\text{semiobserv} + 0.5 \times \text{risks}}{\text{total risks}}
\]

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, semirisks is semi-related risks. Table 2 shows the project risk list.

**Table 2. Project Risk List**

<table>
<thead>
<tr>
<th>Risk ID</th>
<th>Risk name</th>
<th>Nature</th>
</tr>
</thead>
<tbody>
<tr>
<td>R01</td>
<td>Low budget</td>
<td>Cost and time</td>
</tr>
<tr>
<td>R02</td>
<td>Infractions against law</td>
<td>Contract</td>
</tr>
<tr>
<td>R03</td>
<td>Low communication and advertising for the show</td>
<td>User/customer</td>
</tr>
<tr>
<td>R04</td>
<td>Unsuitable cast</td>
<td>Organization</td>
</tr>
<tr>
<td>R05</td>
<td>Unsuitable ticket price-setting</td>
<td>Strategy</td>
</tr>
<tr>
<td>R06</td>
<td>Unsuitable rehearsal management</td>
<td>Controlling</td>
</tr>
<tr>
<td>R07</td>
<td>Cancellation or delay of the first performance</td>
<td>Cost and time</td>
</tr>
<tr>
<td>R08</td>
<td>Poor reputation</td>
<td>User/customer</td>
</tr>
<tr>
<td>R09</td>
<td>Lack of production teams organization</td>
<td>Organization</td>
</tr>
<tr>
<td>R10</td>
<td>Low team communication</td>
<td>Organization</td>
</tr>
<tr>
<td>R11</td>
<td>Bad scenic, lightning and sound design</td>
<td>Technical/performance</td>
</tr>
<tr>
<td>R12</td>
<td>Bad costume design</td>
<td>Technical/performance</td>
</tr>
<tr>
<td>R13</td>
<td>Low complicity between cast members</td>
<td>Technical/performance</td>
</tr>
<tr>
<td>R14</td>
<td>Too ambitious artistic demands compared to project means</td>
<td>Requirements</td>
</tr>
<tr>
<td>R15</td>
<td>Few spectators/lukewarm reception of the show</td>
<td>User/customer</td>
</tr>
<tr>
<td>R16</td>
<td>Technical problems during a performance</td>
<td>Technical/performance</td>
</tr>
<tr>
<td>R17</td>
<td>Low cast motivation</td>
<td>Organization</td>
</tr>
<tr>
<td>R18</td>
<td>Unsuitable for family audiences</td>
<td>Strategy</td>
</tr>
<tr>
<td>R19</td>
<td>Low creative team leadership</td>
<td>Controlling</td>
</tr>
<tr>
<td>R20</td>
<td>Low creative team reactivity</td>
<td>Controlling</td>
</tr>
</tbody>
</table>

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 “phasедone” 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 here as follow is the probability distribution over the two states. \(Pr(s = \text{phaseproceeding}) = 0.50, Pr(s = \text{phasедone}) = 0.50\) where \(s = \text{state at time t}\).
5. 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 Figure 3.


Upon the formulations and descriptions, developed concepts would be combined into a functional framework. The integration process would result a framework which covers complex aspects of SPM training and knowledge acquisition. Figure 4 describes the functional system of conceptual-operative simulation framework for SPM training. As illustrated in figure 4 the framework is completely interactive and captures the decision made by user over the runtime of the simulation process. The decisions outcome would be stored in an element to compute the value of the decision by the decision model which is based on optimal decision policy. This element would be the parameter to determine whether the process concluded with success or with failure.
7. Preliminary Results of the Framework

The main functionality of the framework is determined by POMDP model. To solve the model there is necessity to find an optimal policy. The policy is the foundation for structuring the in-process decision support feature that would map decision with the corresponding action. The determination of decision issues in this paper would be done through the definition of observations in conformity with POMDP model formalism.

For policy determination, the POMDP problem specification file is specified according to the organization of Cassandra [19] definition and the Approximate POMDP Planning Software which was developed at National University of Singapore, school of computing [20], is used. The POMDP input file follows the Tony Cassandra’s format that could be handled by the POMDP solver. It is the formal problem specification files encoding the domain problem under the distinct composition and semantics. Tony Cassandra POMDP specification file must have 5 important objects which specify the discount value, states, actions and observations at the beginning. Figure 5 shows the beginning objects definition for the SPM domain. The order could be in any sequences and all of them must precede specifications of transition probabilities, observation probabilities and rewards.

![Cassandra File Objects](image.png)

**Figure 5. Cassandra [19] File Objects**

In this model there are advantages that would reduce the complexity of finding an optimal policy algorithm: 1-the initial belief point is known, 2-the initial action is
known and 3-belief state transition is one-way in which only transition is from “phaseproceeding” to “phasedone”. These three conditions of the model reduce considerably the complexity of an exponential algorithm. There are 6 actions and 4 observations, according to (3):

\[
\text{Number of policies} = (\text{number of actions})^{\text{number of observations}} = 6^4 = 1296
\]  
\text{It is considerably a large number to find an optimal policy from 1296 existent policies.}

\[
E^\pi(b_0) = \sum_{t=0}^{\infty} \gamma^t E[r(s_t, a_t)|b_0, \pi]
\]  
Where in (4), \(\pi\) is the policy, \(0 < \gamma < 1\) is discount factor, \(r\) is reward function, \(b_0\) is initial belief state and \(E^\pi\) is expected value for policy \(\pi\).

Then the optimal policy would be (5):

\[
\pi^* = \arg\max_{\pi} \left( E^\pi(b_0) \right)
\]

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.

**7.1. Generating Policy**

For the simulated SPM process, the POMDP specification format is designed based on the experience and domain knowledge. Since it is an individual POMDP file, we specify a unique file for each of the user goals and actions. The system can perform 6 types of actions. Number of dialogue states are 2 including the beginning state. The discount value is 0.98 in this experiment. The POMDP solver adopted in this experiment is APPL solver [20]. APPL is a software package which implements several heuristic search algorithms for POMDPs and MDPs. To solve the POMDP problem in our experiments, heuristic search value interaction algorithm (SARSOP) is used.

SAROPS is a point-based approximation algorithm that maintains both upper and lower bounds on the optimal value function, allowing it to use effective heuristics for action and observation selection, and to determine the policy. The following figures shows samples of POMDP file specification for SPM process.

By forming the POMDP specification file in Tony Cassandra's format, the APPL solver produces the out.policy file which specifies each action-state for selected POMDP file along with corresponding approximate optimal solution. In POMDP policy file, a set of lower bound values with an alpha vector and the corresponding actions are presented. With a current belief \(b\), the lower bound on the expected long-term reward starting from \(b\) and that action leading to the expected lower bound can be known. In this experiment, the APPL solver was made to run for 6.10 minutes for the file specified and then stopped for generating the POMDP policy file.

The generated policy file and the original POMDP file are given as inputs to the POMDP solver and the evaluator to make multiple runs and check the number of runs it takes to acquire the result. As well, we can utilize them to validate the POMDP state evaluation which is shown in Figure 6.
Figure 6. Generating Policy from POMDPSPM Binary File

The resulted policy, out.policy file according to APPL POMDP solver is illustrated in Figure 7.

Figure 7. The Resulted Policy According to APPL Solver

In Figure 7, since actions array is commenced from zero, the adjustment of actions is done by adding one to the number in the policy file. The presented elements are alpha...
vectors elicited by the SARSOP algorithm. In each vector row there are two numbers which are accountable for the value of respective action in the space of belief states.

7.2. Determining the Optimal Policy

The optimal policy for the proposed framework is described in Table 3. The transition of belief state with Piecewise linear and convex strategy, is converted into partitions, the belief space (state=phasedone) as represented in Table 3.

Table 3. The Optimal Policy over Continuous Belief State

<table>
<thead>
<tr>
<th>Partition No</th>
<th>Pr(belief space)</th>
<th>Action</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0.000 to 0.100</td>
<td>A1</td>
</tr>
<tr>
<td>2</td>
<td>0.100 to 0.200</td>
<td>A1</td>
</tr>
<tr>
<td>3</td>
<td>0.200 to 0.300</td>
<td>A2</td>
</tr>
<tr>
<td>4</td>
<td>0.300 to 0.400</td>
<td>A5</td>
</tr>
<tr>
<td>5</td>
<td>0.400 to 0.500</td>
<td>A6</td>
</tr>
<tr>
<td>6</td>
<td>0.500 to 0.600</td>
<td>A4</td>
</tr>
<tr>
<td>7</td>
<td>0.600 to 0.700</td>
<td>A6</td>
</tr>
<tr>
<td>8</td>
<td>0.700 to 0.800</td>
<td>A1</td>
</tr>
<tr>
<td>9</td>
<td>0.800 to 0.900</td>
<td>A3</td>
</tr>
<tr>
<td>10</td>
<td>0.900 to 0.999</td>
<td>A3</td>
</tr>
</tbody>
</table>

Table 3 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 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 (PWLC).

7.2.1. Policy Graph of POMDP: Policy graph is another form of 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 8. This graph is over simplified to facilitate a basic understanding of the model functionality. This graph on the other hand provides envisage and clear visual conception for the analyzer to have a better insight on the action, observations and their impact on decision process. Also the 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.
8. Discussion

In-process decision support is realized by application of POMDP technique. This feature is employed by the specified policy of POMDP to evaluate the decision process and calculate decision values. Since POMDP is a stand-alone formalism to formulate the decision process, the policy would be presented as a concrete basis for modeling of the decision process for SPM. 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 decision management system from a decision support system. 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 [21]. 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 automate the process of decision making in a strategic level. 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. With integration of expert systems into this framework, the idea of reaching for having common features of decision support systems and decision management systems will be accomplished.

9. Conclusion

The presented conceptual-operative framework provides different views of SPM training, knowledge management, 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 in which provides tactical view 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.

10. Future Work

The framework is to be transformed into an operational solution. This solution would be constructed upon interactive features of simulation environment complied with SPM training specifications. This framework provides a rich basis for in-process decision 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. Also this feature maintains learning from consequent actions according to the taken decisions instantly.

Another planned 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 process 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 extend another view on the framework. This view is knowledgeable (managerial in implementation) and provides for middle managerial and entails knowledge-based understanding level of the process. This approach also will enable the realization of an exclusive feature to the framework to operate on knowledge development dimension of knowledge management process.

Acknowledgements

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. The financial of this project is supported by Ministry of Higher Education Malaysia and Universiti Teknologi Malaysia under Vot No: 07J87.

References


Authors

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.