The Architecture of an Effective SOFTWARE APPLICATION FOR MANAGING ENTERPRISE PROJECTS

1. INTRODUCTION

To complete any enterprise project, a large collection of interrelated activities should be accomplished. These activities are performed through utilizing a diverse set of resources including people, finance, equipment, and materials which need to be managed on the daily basis. This implies that managing enterprise projects is mainly concerned with optimizing the allocation of such resources to a variety of tasks in different time slots, and possibly different geographical locations. It is worth noting that the objective function of resource optimization should definitely reflect the intelligent trade-offs between cost, duration, safety, and quality of the project. Overviews on project baseline scheduling and project data for integrated project management have been provided in (Vanhoucke, 2013) and (Vanhoucke, Coelho, & Battelier, 2016), respectively.

Since an enterprise project includes a large number of activities and each activity has a series of precedence and resource constraints, four interrelated factors make the management of such projects a challenging task: (i) different types of precedence constraint indicating how an activity can start with respect to its predecessors, (ii) various types of resource constraints with respect to limiting the set of activities that can be in process at the same time, (iii) enormous variety of limited resource types, (iv) a variety of conflicting factors like cost, quality, and time affecting the implementation of the activities based on their different execution modes. It is the combination of these four factors which necessitates the implementation of an effective decision making process, full of different optimization techniques, and capable of making intelligent trade-offs.

Introduction and utilization of information systems, as a strategic tool in business, can usually facilitate decision making processes through an array of intelligent trade-offs, ranks unused hit marks and use them in appropriate time-slots. By hit marks here we mean valuable resources that in some time-slots of the project can increase the mark of performance in terms of cost, quality, time, or security. Hence, the presented architecture has been called Multi Agent RSH Architectural Layout (MARSHAL). Figure 1 shows the components of the MARSHAL.

For a given enterprise project, a variety of optimization objectives are possible, such as minimizing project duration; maximizing net present value; minimizing delay; and maximizing resource utilization. In general, however, the manager should perform the allocation of different types of resources to different activities of the project to achieve a proper trade-off in four major concerns of cost, time, safety, and quality.

The advance of technology, in general, and that of communication technology in particular, along with the convergence of computation and communication, affects how project managers handle this trade-off (Gutierrez & Friedman, 2005; Jaffari & Maniowong, 1998; Raymond & Bergeron, 2008; Wateridge, 1999). A set of principles for performing mixed methods research in IS has been introduced in (Venkatesh, Brown, & Bala, 2013), and selection criteria for strategic project design has been deliberated in (Benedetto, Bernardes, & Vieira, 2016). In effect, in the recent view, project man-
The structure of the paper is as follows: Section 2 discusses how the supports in the key areas of architecture, providing managers with all sophisticated decision making tools for the project management. The MARSHAL’s design is based on the availability that accompanies the scheduling phase. Changes from single, to multi-mode scheduling (Bouleimen & Lecocq, 2002), and (Zamani, 2011). All the procedures mentioned produce solutions integrating both exact and heuristic methods include those presented by (Sprecher, Bhattacharya, & Bagchi, 1999), and (Zamani, 2010b). Solutions integrating both exact and heuristic methods include those presented by (Sprecher, 2002), and (Zamani, 2011). All the procedures mentioned produce schedulers that are utilized for proper project management.

General discussions about blended learning and the application of operations research in project scheduling have been presented in (Vanhoucke, Knust, Schoo, & Thiele, 1998), (Zamani & Shae, 1998), (Nazareth, Verma, Bhattacharya, & Bagchi, 1999), and (Zamani, 2010b). Solutions integrating both exact and heuristic methods include those presented by (Sprecher, 2002), and (Zamani, 2011). All the procedures mentioned produce schedulers that are utilized for proper project management.

General discussions about blended learning and the application of operations research in project scheduling have been presented in (Vanhoucke, 2014). When activities can be performed in different modes, the problem changes from single, to multi-mode scheduling (Bouleimen & Lecocq, 2003; Mori & Tseng, 1997; Tereso, Araujo, & Elmaghraby, 2004; Zamani, 2013b), and when the duration of activities is not exact, the problem changes from a deterministic model to a stochastic one (Tereso et al., 2004). In (Hutchings, 2004) the three major tasks of planning, organizing, and controlling are considered as operating systems for all schedules. In effect, these operational systems are the major bases of project management that accompany the scheduling phase.

MARSHAL focuses on how information systems play a key role in making intelligent trade-offs, deploying principles, insights and techniques, which we characterize by the term, Project Management Information Systems (PMIS). The MARSHAL’s design is based on the fact that despite the availability of various approaches for dealing with the key areas of planning, organizing, and controlling activities of projects, there are still many challenges ahead and placeholders need to be envisaged in the corresponding management architecture, providing managers with all sophisticated decision making supports in the key areas.

The structure of the paper is as follows: Section 2 discusses how the MARSHAL can be considered as a set of intelligent problem solving techniques, and Section 3 examines principles, insights and techniques employed in the presented architecture. Section 4 describes the interaction of concepts used in MARSHAL with one another; and Section 5 discusses the relationship of traditional areas of project management with the presented architecture. A template software application is presented in Section 6, and concluding remarks are discussed in Section 7.

2. MARSHAL AS A SET OF INTELLIGENT PROBLEM SOLVING TECHNIQUES

MARSHAL focuses on how effective communication among various agents which collaborate to execute a project, occurs with minimum interaction, through efficient coordination. The complexity of enterprise project management stems from its reliance on many types of resources and diverse technical expertise, all of which need to be formally coordinated through instruments such as contracts and schedules. The role of information systems to effectively facilitate these interactions and provide a concrete record of such interactions in ways that satisfy the requirements of legal, finance, accounting, engineering and computer science, as representative examples, is of critical importance.

An effective communication channel for all participants of the projects has a key importance in updating databases needed for these trade-offs. The rapid advance of communication technology reflected in the fast services provided by the Internet, highlights the advancements made in computation technology, and the complexity involved in managing projects, which is mainly the result of a wide types of scarce resources needed to accomplish each of many activities comprising the project. These principles and insights are aimed at facilitating appropriate infrastructure for an effective software architecture.

With respect to employing these principals and insights, it is worth emphasizing that software architecture is a very broad term and we have used it to denote all the preparation needed between the two stages of gathering requirements and preparing a detailed design for writing the full executable computer code. Towards this direction, MARSHAL uses an innovative Tier-Layer Principle (TLP), which is aimed at decreasing vulnerability of the PMIS to dynamic change of requirements. In effect, the TLP makes the implementation of the PMIS possible by facilitating the dropping of any current feature and adding any extra one, at any time such a current feature is not needed or the extra feature is required.

The TLP works based on two complementary facades of layers and tiers, with layers handling cohesion and tiers handling the processing boundary. With regard to layers, the classes needed to develop a PMIS software application are grouped in several layers, with functionally or logically related classes being located in the same layer. Cohesion between the classes of the same layer and the distribution of proposed system functionality among the classes are the main concerns of this layerd architecture. The coupling of any class to other class is kept minimized and the classes are kept as independent as possible. By ‘independence’ we mean that the relationship between two classes is such that when a change occurs in one class, that change does not affect the other class. Figure 2 shows the proposed layers.

3. PRINCIPLES AND INSIGHTS EMPLOYED IN THE DEVELOPMENT OF MARSHAL

Based on effective utilization of the Internet, MARSHAL has been based on disseminating the right information to the right people at the right time and in the right format, providing the infrastructure needed to make informed decisions about finance, planning, scheduling, and procurement of an enterprise project. Three points highlight the importance of the principles and insights employed in MARSHAL: (i) the enormous advances of Internet technology, (ii) the growth of the decision making industry as a result of advances made in computation technology, and (iii) the complexity involved in managing projects, which is mainly the result of a wide types of scarce resources needed to accomplish each of many activities comprising the project. These principles and insights are aimed at facilitating appropriate infrastructure for an effective software architecture.

With respect to employing these principals and insights, it is worth emphasizing that software architecture is a very broad term and we have used it to denote all the preparation needed between the two stages of gathering requirements and preparing a detailed design for writing the full executable computer code. Towards this direction, MARSHAL uses an innovative Tier-Layer Principle (TLP), which is aimed at decreasing vulnerability of the PMIS to dynamic change of requirements. In effect, the TLP makes the implementation of the PMIS possible by facilitating the dropping of any current feature and adding any extra one, at any time such a current feature is not needed or the extra feature is required.

The TLP works based on two complementary facades of layers and tiers, with layers handling cohesion and tiers handling the processing boundary. With regard to layers, the classes needed to develop a PMIS software application are grouped in several layers, with functionally or logically related classes being located in the same layer. Cohesion between the classes of the same layer and the distribution of proposed system functionality among the classes are the main concerns of this layerd architecture. The coupling of any class to other class is kept minimized and the classes are kept as independent as possible. By ‘independence’ we mean that the relationship between two classes is such that when a change occurs in one class, that change does not affect the other class. Figure 2 shows the proposed layers.

Classes are distributed in three tiers. These tiers are (i) model, (ii) view, and (iii) controller. The selection of tiers is based on the guidelines of the MVC (Model, View, and Controller) architecture (Gurigallu, 2014). A class is included in the view tier if it is responsible for showing any information to or managing any interaction with users, and is included in the model tier if it handles the data sources of projects. The third tier, controller, includes classes responsible for computation and connecting the view with model. Figure 3 shows how in our proposed architecture, model, view and controller tiers interact with one another.

Hence, in our multi-agent approach, every agent is represented with (l, t, i) in which the indices l and t show the corresponding layer and tier, and index i has been used to differentiate agents having the same l and t. In the case there is only one class in a tier l and t, the value of i for that specific values of l and t is only 1. By relating layers to tiers, Figure 4 depicts the foundation of (l, t, i) notation for accessing agents.
MARSHEL is aimed at providing both the effective sharing of information among a team of experts handling a project and the provision of intelligent solutions. To manage the complexity of resources to activities towards fulfilling the goals of the project.

The implementation of a PMIS is highly involved with eight distinct concepts: (i) Intelligent Systems, (ii) Systems Simulation, (iii) Mathematical Programming, (iv) Stochastic Processes, (v) Database Systems, (vi) Requirement Engineering, (vii) Systems Analysis, and (viii) User Interface and Usability. Figure 5 shows how in MARSHEL these concepts interact with one another.

Because the backbone which interconnects these areas is the Internet, the second principal employed in the development of MARSHEL is the effective use of the Internet. This principle highlights the importance of disseminating various types of information for facilitating the communication of all people involved in the project through accessing to the right information at the right time and with the right format, from different geographical locations. As well as being the backbone of the above interconnections, the Internet can also enhance the management of traditional areas of project management.

The second area, human resource management, is involved with the most effective utilization of people involved in a project. The major approach suggested for dealing with this area are Assignment Models. These models are associated with assigning sets of resources to sets of activities. This is done such that all resource limiting constraints, as well as the accomplishment of all scheduled activities, are satisfied whilst minimizing cost. In order for assignment models to be useful in the PMIS, they should be able to handle all the soft and hard constraints associated with allocation of resources.

With respect to enterprise projects, these models should, in particular, be capable of handling a set of dynamic objective functions. Third, communication management is about ensuring the suitable generation, assimilation, storage, and distribution of project information. Without proper communication management, an enterprise project cannot be successful. That is why in regard to establishing such management, Electronic Data Interchange (EDI) models are suggested to be the base of advanced software applications developed (Tan, Liu, Li, & Zhao, 2014). The base of EDI models is that, in order for data to be shared, the users should have the same understanding from the same piece of data. Therefore, these models are intended to remove potential ambiguities.

By providing standard definitions for words, and using standard formats for communication, the experts involved in the PMIS can communicate effectively and efficiently. Despite the importance of EDI in many different fields, to the best of our knowledge, no significant effort has been made in the literature to highlight its role in PM.

Fourth, project quality management in enterprise projects is a key to guaranteeing high performance, with its main principle being to satisfy the needs for which the project has been undertaken. In this regard, Quality Performance Index (QPI) models can be considered the major principle on which effective software applications can be developed (Engemann, 2014). By systematically categorizing the factors affecting quality, these models provide facilities for defining, planning and controlling different qualities expected in different parts of the project.

Fifth, the cost of any project is a key concern in its execution, and for enterprise projects, managing cost, because of it high volume, is of high significance. Proper cost management ensures that an enterprise project can be accomplished within an acceptable threshold of the approved budget. With respect to an MARSHEL perspective, Network-Based Cost-Benefit Analysis models (McReynolds, Lawrence, & Pajet, 2013) can be considered a major principle on which effective software applications can be developed for managing the cost of enterprise projects.

As their name reveals, these models base their analysis of costs and benefits on networks. By considering networks as the base of analysis, all precedence relations between activities of the project, as well as all the availabilities of resources associated with these activities can be taken into account to manage cost as effectively as possible. Despite the importance of these models in many different fields from telecommunication to system engineering, to the best of our knowledge, there is no article describing their uses in PM.

Sixth, risk is an inevitable part of any project, in general, and of enterprise projects in particular. Managing risk in enterprise projects is, however, much more challenging than in ordinary projects as different hazards propose different levels of risks to the various stakeholders. Risk management in enterprise projects should precisely identify and analyze possible risks throughout the life of the project and handle these risks most effectively. In this regard, Advanced Stochastic Process Models (Pearl, 2000) are proposed to be the base of the software applications developed for this purpose.

These models can be used to investigate the behavior of projects and provide the management team with the appropriate strategies to mitigate the risks.
of interrelated variables interacting with one another to execute a project. The importance of stochastic models in project management has also been highlighted in (Hutcheson, 2004).

Seventh, enterprise projects are highly involved with procurement, and managing such procurement deals with proper outsourcing, in the sense of suitable acquiring of goods and services needed by the project from outside. E-Commerce Revenue Models (Mahadevan, 2000) are proposed as the major principal needed for developing software applications for this purpose. From MARSHAL perspective, the rationale behind this proposal is that the project procurement management is highly involved with distributed decision making, which necessitates proper flow of information.

Eights, perhaps after the cost, the time of any project is the second key concern in its execution, and for enterprise projects, managing time is of high significance. For enterprise projects, time management is necessary to secure the accomplishment of the project within an acceptable time threshold. In many cases, the timely completion of enterprise projects is perhaps the single most critical issue considered in their execution. In this regard, Probabilistic Time-Resource Estimate Models are proposed as a base for the software application developed. Several of these models have currently been implemented and their results are promising (Hutcheson, 2004; Love & Irani, 2003; Panagiotakopoulos, 1977). By using these stochastic models, making intelligent tradeoffs between the cost of resources used and the completion time of the project as well as the performance of the project becomes possible. After all, time, cost, and performance affect each other and are not independent variables.

All of the eight proposed principles are in the direction of organizing information and facilitating communications to make effective choices among different alternatives. In effect, with respect to MARSHAL perspective, information, communication, and intelligence can be considered as the main factors affecting the components of the PMIS. Based on these three factors, MARSHAL effectively supports all of the planning, scheduling, and controlling phases. Whereas Figure 6 shows the role of planning, scheduling, and controlling, Figure 7 shows the interplay of Information, Communication, and Intelligence.

As is shown in Figure 7, MARSHAL is aimed at dynamically identifying the bottlenecks constraints and breaking them. The term ‘dynamically’ is here used because, after breaking an identified constraint, another constraint becomes a bottleneck and should be broken in turn. The chain of identifying and breaking bottlenecks continues until all activities are accomplished. Consider- ing the importance of control in the management of an enterprise project, the creation of such an effective chain by MARSHAL can highly impact the overall performance of the corresponding procedure.

6. A TEMPLATE SOFTWARE APPLICATION

A Template software application has been programmed in a combination of C++ and Visual Basic Application for Excel (VBA EXCEL). While its C++ component uses an evolutionary search technique taken from (Zamani, 2013b) and modified to suit this environment, the interface component, which has been programmed in Visual Basic Application for Excel (VBA EXCEL) and uses EXCEL capabilities in providing the necessary interface. Based on the MVC principal of MARSHAL, a controller component has also been considered. These three components interact with one another through Component Object Model (COM) provided in the windows operating system. In this way, the template can use all the capabilities of EXCEL both in interacting with the user and in presenting its graphical outputs.

Figure 8 shows the backbone of the template and Figure 9 shows the feedback process in balancing the cost and duration of the project in the template. The template retrieves the information of the project from its data store (Model) and displays it at the request of users in the interface (View). Users usually change the data, and template needs to store the changes in its data store. The point is that there is no tie between the interface and data store. In fact the coupling of the data and user interface pieces has been replaced with a piece (Controller) which incorporate project management logic exceeding far beyond data transmission between data store and interface. In MARSHAL, this project management logic can include hundreds of sophisticated optimization techniques, which in the current template are not present.

Deeper issue is that, at a given level of safety, shortening the time and decreasing the cost can downgrade the quality of the project whereas increasing quality and decreasing time usually leads to higher cost. MARSHAL has been designed for dealing with such complexities. In the design of MARSHAL it has been noticed that in enterprise projects, making trade-off among conflicting factors is not trivial and requires the use of sophisticated techniques and proper principle incorporated in the software application managing the project and making delicate trade-offs.

In developing MARSHAL, MAS has been used to provide principles and insights needed for the design of a proper information system dealing with decision making complexities through using computing and communication technologies.

In effect, by the convergence of computing and communications, these methods, principles, and insights can be viewed as hierarchically related components creating the base of a software application aimed at facilitating the flow of information in organizations to make intelligent trade-offs possible. MARSHAL, as a web-based architecture, presents such a hierarchically related components towards facilitating decision making process through integrating, storing, editing, sharing, and, most importantly, making intelligent trade-offs among time, cost, and quality.

Considering a large array of trade-offs needed to be made for the accomplishment of enterprise projects, the importance of MARSHAL can be highlighted by the diversity of the types of scarce resources which are needed by activities and the broad range of parameters needed to be taken into account usually through the Internet, in order to execute such projects. MARSHAL can handle both (i) the direct problems like maximization of the performance quality for a given cost, safety level, and duration and (ii) the inverse problems like minimizations of cost, or duration, for a given performance quality and safety level. Moreover relating to layers in developing the proposed (L.1) foundation for accessing agents in the proposed multi-agent approach has the potential of leading to a highly effective design, needed for the full computer coding of the proposed software application.

7. CONCLUSIONS

MARSHAL deals with decision-making complexity inherent in enterprise projects through emphasizing on computation, communication, and optimization in a multi-agent environment, best suited for super computers with massive parallel processing capability. Ability of coping with different objective functions reflecting different trade-offs needed for conflicting factors is the prime consideration in the architecture presented.

In effect, managing enterprise projects is involved with an array of complex decision making tasks. The issue is not simply that in decision making process of a project, a wide range of variables like time, cost, and quality, interplay with one another. The
Chassan Kreyden is a Professor of Information Systems at the School of Systems and Management and Leadership at University of Technology, Sydney. He has received a degree in computer engineering and a PhD in knowledge systems from the University of New South Wales. His research interests include computer systems modeling, distributed decision support systems and knowledge analyzes. He has been applying his results in developing decision support systems for disaster management, risk management systems, a learning system, and others. He is currently working on a project sponsored by an Australian Research Council Grant to investigate the best ways to design knowledge in developing methodologies for complex systems and the same time working on decision support systems with SES on exploring the use of ontologies for flood management decision support. He has authored more than 100 journal and conference papers in these areas over the past 15 years. His most recent publication appeared in Information and Management Systems (IEMS), IEEE Transactions on Knowledge and Data Engineering (TKDE), Information Systems Journal (ISJ), Journal of Systems and Software (JSS), International Journal on Artificial Intelligence Tools (IJAIT) and Journal of Computer Information Systems (JCOM). He also serves on the editorial boards of several international journals including: International Journal of Intelligent Information Technologies (IJIT), Journal of Software and others.

Ghassan Beydoun is a Professor of Information Systems at the School of Systems, Management and Leadership at University of Technology, Sydney. He received a degree in computer engineering. Ghassan’s research includes: Computer systems modeling, distributed decision support systems and knowledge analytics. He has been applying his results in developing decision support systems for disaster management, risk management systems, a learning system, and others. He is currently working on a project sponsored by an Australian Research Council Grant to investigate the best ways to design knowledge in developing methodologies for complex systems and the same time working on decision support systems with SES on exploring the use of ontologies for flood management decision support. He has authored more than 100 journal and conference papers in these areas over the past 15 years. His most recent publication appeared in Information and Management Systems (IEMS), IEEE Transactions on Knowledge and Data Engineering (TKDE), Information Systems Journal (ISJ), Journal of Systems and Software (JSS), International Journal on Artificial Intelligence Tools (IJAIT) and Journal of Computer Information Systems (JCOM). He also serves on the editorial boards of several international journals including: International Journal of Intelligent Information Technologies (IJIT), Journal of Software and others.

References


Dr. William Turley is currently a Lecturer in the School of Computing and Information Technology. Before joining the university in 2000, William worked in project management with Australian Broadcasting Corporation and technical training roles throughout the South Pacific region. In his PhD William investigated Community Technology Centers in Australia to review public policy, resources aimed to overcome the digital divide. He has also completed a Masters by Research which looked into the challenges faced by broadcast technicians in the digital era. He has completed a number of projects involving corporate training for large corporate networking and management and other. He serves on the board of a number of international journals including: International Journal of Intelligent Information Technologies (IJIT), Journal of Software and others.

Dr. William Turley is currently a Lecturer in the School of Computing and Information Technology. Before joining the university in 2000, William worked in project management with Australian Broadcasting Corporation and technical training roles throughout the South Pacific region. In his PhD William investigated Community Technology Centers in Australia to review public policy, resources aimed to overcome the digital divide. He has also completed a Masters by Research which looked into the challenges faced by broadcast technicians in the digital era. He has completed a number of projects involving corporate training for large corporate networking and management and other. He serves on the board of a number of international journals including: International Journal of Intelligent Information Technologies (IJIT), Journal of Software and others.