PROJECT TRADE-OFFS

The Architecture of an Effective

SOFTWARE APPLICATION FOR MANAGING ENTERPRISE PROJECTS

DR. REZA ZAMANI
· Senior Lecturer, School of Computing and Information Technology, EIS
· University of Wollongong, Australia
· reza@uow.edu.au

DR ROBERT B. K. BROWN
· Student, School of Law
· University of Wollongong, Australia
· rbkb788@uowmail.edu.au

PROFESSOR GHASSAN BEYDOUN
· Deputy Head of School, School of Systems, Management, and Leadership
· University of Technology, Australia
· ghassan.beydoun@uts.edu.au

DR. WILL TIBBEN
· Lecturer, School of Computing and Information Technology, EIS
· University of Wollongong, Australia
· wjt@uow.edu.au

· ABSTRACT ·

This paper presents the architecture of an effective software application for managing enterprise projects. Viewing the execution of an enterprise project as a highly complex system in which many delicate trade-offs among completion time, cost, safety, and quality are required, the architecture has been designed based on the fact that any action in one part of such a project can highly impact its other parts. Highlighting the complexity of the system, and the way computational intelligence should be employed in making these trade-offs are the base of the presented architecture. The architecture is also based on the fact that developing a software application for appropriate managing of such trade-offs is not a trivial task, and a robust application for this purpose should be involved with an array of sophisticated optimization techniques. A multi-agent system (MAS), as a software application composed of multiple interacting modules, has been used as the main component of architecture. In this multi-agent system, modules interact with environment on-line, and resolve various resource conflicts which are complex and hard-to-resolve on daily basis. Based on the proposed architecture, the paper also provides a template software application in which an array of optimization techniques show how the necessary trade-offs can be made. The template is the result of the integration of several highly sophisticated recent procedures for single and multimode resource-constrained projects scheduling problems.

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 geographic 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, & Batselier, 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 or 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 complicated components (Xu, Wijesooriya, Wang, & Beydoun, 2011). Among these components, Multi-Agent Systems (MAS) and Operations Research (OR) play key roles, with MAS comprising software applications composed of multiple interacting modules, and OR being part of the interdisciplinary field of management science, based on mathematical modeling and simulation to assist in identifying optimal trade-offs.

Agent-based modeling compliments the design of MAS. Whereas MAS tackles complexity, agent-based models (ABM) represent the problem appropriately to enable a MAS solution. ABM results from an agent oriented analysis of the problem, where autonomous components are allocated localized sub-problems to solve in a divide-and-conquer approach (Beydoun, Tran, Low, & Henderson-Sellers, 2006). In this manner, MAS decomposes complexity and solves the sub-problems, thus providing a solution to the overall problem (Beydoun, Low, Tran, & Bogg, 2011).

ABM provides a detailed problem description in terms of the behavior of interacting components (agents). It further generates an explanatory description of control within the corresponding multi agent system. Thus, ABM facilitates an abstract description of MAS. Hence, in architecting a multi-agent system, it is mainly AMB which provides a detailed description in terms of the behavior of interacting components. In this paper, however, for the sake of simplicity, we assume that ABM typically precedes the construction of a MAS, thus we allow for the term MAS to subsume ABM, and we will simply refer to as MAS.

The design of the architecture presented in this paper has been based on a "Ranking Spare Hit marks" (RSH), in the sense that this architectural layout, in making intelligent trade-offs, ranks unused hit marks and use them in appropriate timeslots. 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.
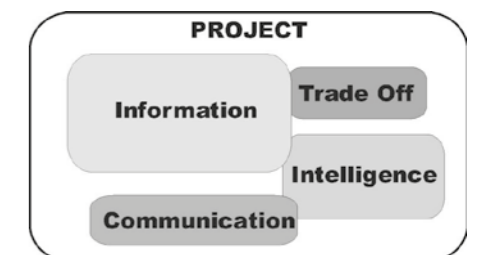

FIGURE 01. The components of MARSHAL

For a given enterprise project, a variety 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 the way project managers handle this trade-off (Gutierrez & Friedman, 2005; Jaafari & Manivong, 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-

agement is seen as an integrated approach towards planning, scheduling, and controlling (Kerzner, 2013). This is highly involved with the incorporation of intelligence into various procedures needed for project management. It is worth noting that the integration of intelligence into MARSHAL is not limited to specific data structures or algorithmic approaches. Indeed, a variety of implementation approaches, ranging from systems simulation, thorough mathematical programming, to approaches based on collective intelligence are employed.

The literature of project scheduling, as a major technical part of project management, dates back to the 1950s and since then different procedures have been proposed for the problem. In general, these procedures are divided into heuristics and exact methods. Although, as mentioned above, numerous objective functions may apply to the management of projects, a majority of these procedures have been proposed for minimizing project duration. Extensive literature surveys on these procedures have been presented in (Özdamar & Ulusoy, 1995), (Hartmann & Kolisch, 2000), (Kolisch & Hartmann, 2006). Moreover, several agent-based solution methodologies have been developed towards solving this problem (Agarwal, Tiwari, & Mukherjee, 2007; Jedrzejowicz & Ratajczak-Ropel, 2007; Zamani, 2010a; Zamani, 2013a). Effective exact solutions for the problem have also been proposed in (Demeulemeester & Herroelen, 1992, 1997), (Brucker, Knust, Schoo, & Thiele, 1998), (Zamani & Shue, 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 schedules 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 are not exact, the problem changes from a deterministic mode 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 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 the eight 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 only one aspect of such a channel. It is apparent that advances in decision making software applications underpinned by the speed of computation technology outlines a role for multi-agent decision making in the presented architecture.

MARSHAL is the integration of sophisticated information systems with PM, aimed at formulating effective decision making models by optimization packages. Since MARSHAL performs the allocation of resources to interrelated activities with consideration given to factors such as time, cost, safety, environment, and quality; it can be viewed as using information technology for facilitating decision making through intelligent trade-offs among the conflicting factors of time, cost, safety, and quality. Considering the fact that every project is a dynamic entity with its own unique purpose

that requires specific resources and should be accomplished in a specific time, MARSHAL is aimed at facilitating the provision of a sophisticated flow of information among the members of a team of experts in relation to making intelligent trade-offs about the effective accomplishment of the project.

Using a diverse set of databases and optimization packages, MARSHAL is also aimed at maximizing an objective function reflecting delicate trade-offs mentioned. In this regard, the MARSHAL presents a set of intelligent problem solving techniques facilitating planning, scheduling and controlling various phases of a project. Considering that PM is involved with the four major conflicting factors of time, cost, safety, and quality, MARSHAL has to make the effective trade-offs needed by using information, intelligence, communication, and feedback.

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 facets of layers and tiers, with layers handling cohesion and tiers handing 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 layered 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.
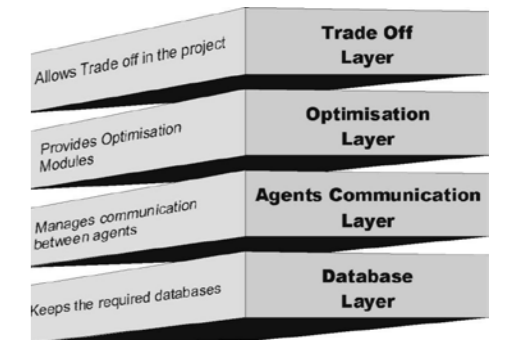
FIGURE 02. Coping with vulnerability of the PMIS to dynamic change of requirements through introducing four separate layers in MARSHAL

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 indexes 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 agent having index 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.
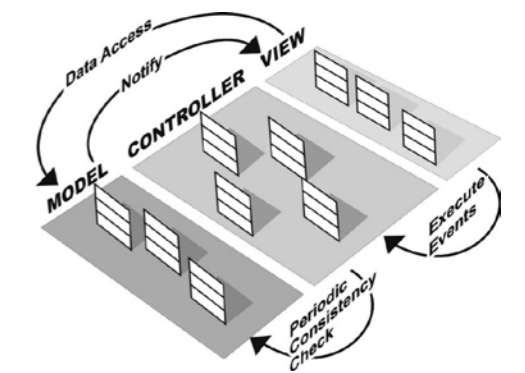
FIGURE 03. Coping with vulnerability of the PMIS to dynamic change of requirements through introducing three separate intersecting tiers in MARSHAL
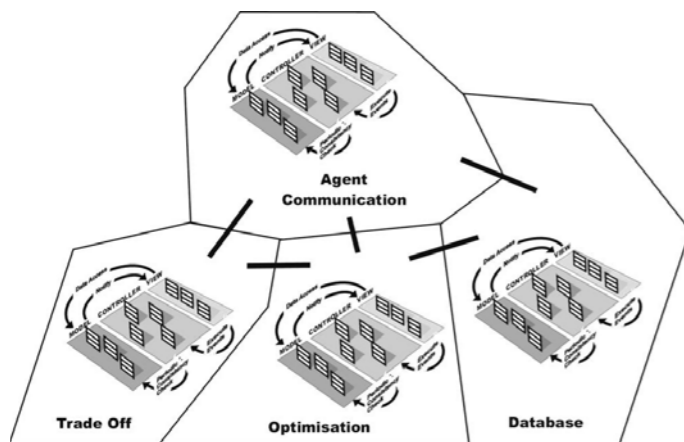
FIGURE 04. Relating tiers to layers in developing (l, t, i) notation for accessing agents in the proposed multi-agent approach employed in MARSHAL

4. THE INTERACTION OF CONCEPTS USED IN MARSHAL WITH ONE ANOTHER

MARSHAL 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 architecting a PMIS, a model driven approach as advocated in (Beydoun and Low 2013) is followed. In this regard, MARSHAL is viewed as the integration of six highly interrelated concepts:

(i) Different communication and optimization procedures involved in the management of a project along with the set of facilities that, in improving decision making process, each provides necessary assistance.

(ii) The interdependency of the procedures in providing information for intelligent trade-offs, facilitated through the Internet or an Intranet, and aimed at effectively supporting decisions made with respect to the optimal utilization of resources.

(iii) Flow of information in providing procedures for making intelligent trade-offs needed for managing a project.

(iv) Contact points comprising the interface of the software with a variety of possible users located in different geographical locations.

(v) The format and the type of possible input entries provided by different users through the contact points.

(vi) The format and the type of possible output generated by the software application.

By considering the six concepts outlined above, the application performs the three main tasks of: (i) providing effective solutions in regard to planning, organizing, and controlling of activities (ii) receiving the information about resource con-

straints from the environment through a web-based system, (ii) optimizing the objective function of the project subject to the constraints received on-line. Producing these solutions requires integration of information, intelligence, and feedback. This integration is based on four concepts: (i) system analysis, which provides effective understanding of the environment of the project, (ii) system engineering, which views a project as systems of interacting components performing within the environment of the project; (iii) system planning, which addresses technological issues related to the planning of the project, and (iv) system scheduling, which comes after the system planning phase, and is related to the optimal allocation 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 MARSHAL these concepts interact with one another.

Because the backbone which interconnects these areas is the Internet, the second principal employed in the development of MARSHAL 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.
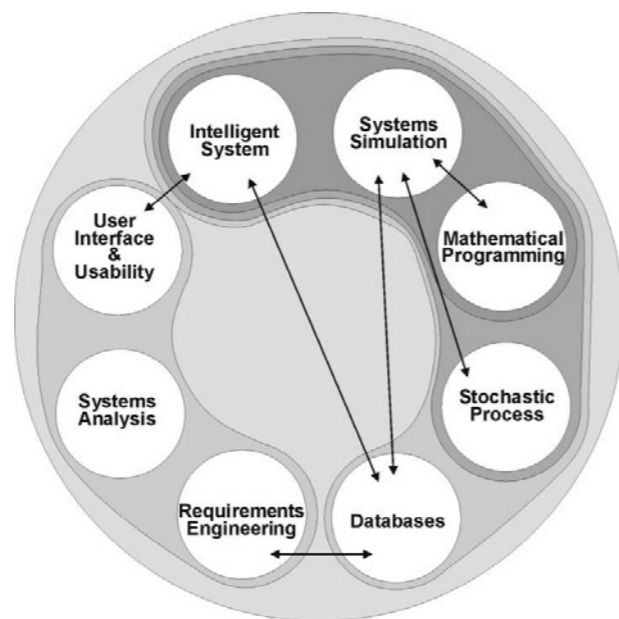

FIGURE 05. The interaction of concepts used in MARSHAL with one another

5. MARSHAL WITH RESPECT TO THE EIGHT TRADITIONAL AREAS OF PROJECT MANAGEMENT

As a major authority which has thousands of project managers as members, the Project Management Institute categorizes the traditional areas of project management as: (i) scope, (ii) human resources, (iii) communication, (iv) quality, (v) cost, (vi) risk, (vii) contract/procurement, and (viii) time. In each of eight areas, enterprise projects need extra services and the Internet can facilitate the corresponding services. These extra services on which MARSHAL has been designed are as follows.

First, without a proper scope, an enterprise project cannot exactly define what is and what is not included in the project, and with respect to determining such a scope, logical requirements models are the base for the advanced software application developed. Without specifying the scope of a project, nothing can be further formulized about it, making the determination of a proper scope for any enterprise project of paramount importance.

Scope definition should be implemented through a systems analysis approach and Logic Requirement Models, as a major principle associated with scope management, play a key role. By using these models, the characteristics of the project are specified through the interaction of a computer program with a diverse set of experts. These models, the bases for intelligent software applications, can be established using state-space search with logical variables (Shimbo & Ishida, 2003).

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 recourses (P.M. Pardalos & Pitsoulis, 2000; P.M. Pardalos , Rendl, & Wolkowicz, 1994). 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, assortment, 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 forms 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 MARSHAL perspective, Network-Based Cost-Benefit Analysis models (McReynolds, Lawrence, & Pujet, 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 that in ordinary projects as different hazards propose differing levels of risks to the various stakeholding participants. 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 interrelated random variables interacting with one another to execute a project. The importance of stochastic models in project management has also been highlighted in (Hutchings, 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 the fact that 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 (Hutchings, 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 bottleneck 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 bottleneck constraints continues until all activities are accomplished. Considering 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
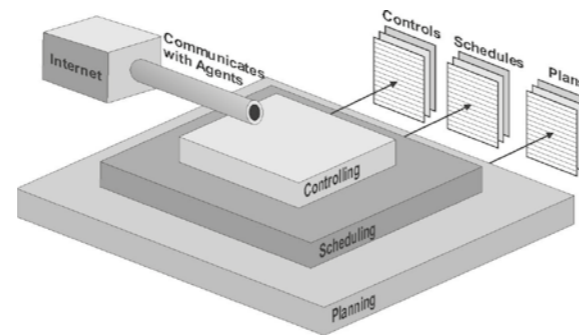

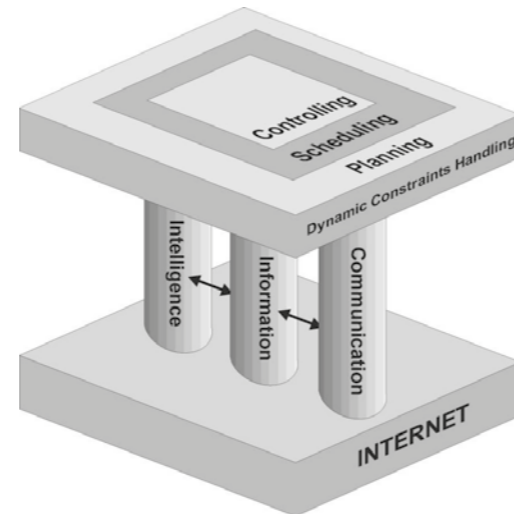FIGURE 06. The Role of Planning, Scheduling, and Controlling in MARSHAL

FIGURE 07. The interplay of Information, Communication, and Intelligence in MARSHAL

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 ballancing 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 inteface. In MARSHAL, this project managemnt logic can include hundreds of sophisdtacted optimization techniques, which in the current tempalte are not present.
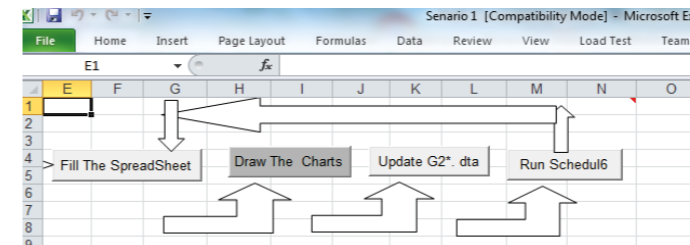

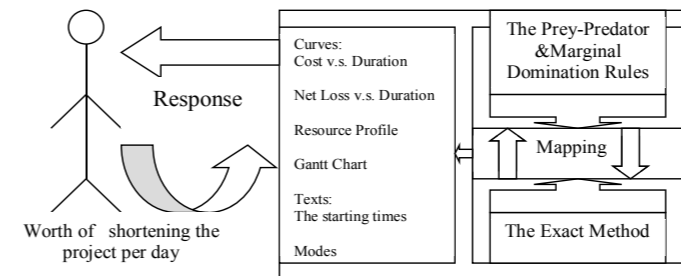FIGURE 08. A Template Software Application Based on VBA Programming in EXCELL

FIGURE 09. The feedback process in ballancing the cost and duration.

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, safety, and quality interplay with one another. The

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 shortening 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 an array 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 utilizing 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, quality and safety.

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 experts needed to communicate with one another, 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 tiers to layers in developing the proposed (l, t, i) notation 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.　◆

• AUTHORS •

DR REZA ZAMANI has received his PhD degree from University of Wollongong, Australia, in 1995 and currently is a senior Lecturer in the School of Computing and Information Technology of the same university. His research interests include project scheduling, optimization, and evolutionary computation. The results of his research in project scheduling have been published in journals like IEEE Transactions of Evolutionary Computation, Evolutionary Computation (MIT Press), and European Journal of Operational Research.
LAST PUBLICATIONS:
1. M Amirghasemi, and R. Zamani (2017) An effective evolutionary hybrid for solving the permutation flowshop scheduling problem, Evolutionary computation (MIT Press) Vol 25, No 1, 87-111.
2. R. Zamani (2017) An evolutionary implicit enumeration procedure for solving the resource-constrained project scheduling problem, International Transactions in Operational Research To be Published in Volume 25.
3. S.K. Lau, R. Zamani, W. Susilo (2016) A Semantic web vision for an intelligent community transport service brokering system, IEEE Intelligent Transportation Engineering, Singapore, IEEE DOI: 10.1109/ICITE.2016.7581328.

DR. ROBERT B. K. BROWN, after a brief initial career researching and teaching in the History and Philosophy of Science and some period developing and supporting IT systems in industry, was awarded a PhD in Computer Science for developing an application of Activity Theory to the conversion of Business Processes into implementable Systems Development requirement specifications (the ATSA Method). He lectured and researched in Systems Development Methodologies, Project Management, Databases, Algorithms, Professional Ethics and related topics at the University of Wollongong for several years before commencing a career in the Law.

• AUTHORS •

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 and a PhD degree in knowledge systems from the University of New South Wales. His research interests include complex systems modelling, distributed decision support systems and knowledge analysis. He has been applying his results in developing decision support systems for disaster management, multi agent systems, e-learning systems and other areas. He is currently working on a project sponsored by an Australian Research Council Discovery Grant to investigate the best uses of domain knowledge in developing methodologies for complex systems and at 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, Information Systems Frontiers, IEEE Transactions of Software Engineering, Information Systems journal, International Journal of Human Computer Studies, Information Processing and management and others. He serves on the board of a number of international journals including: International Journal of Intelligent Information Technologies (IJIIT), Journal of Software and others.

DR. WILLIAM TIBBEN 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 South Pacific region. In his PhD William investigated Community Technology Centers in Australia to review public policy measures aimed to overcome the digital divide., He has also completed a Masters by Research which looked into the challenges faced by broadcast technicians in isolated locations of the Pacific. William has been a member of the IEEE for over twenty years and his teaching responsibilities include project management subjects in corporate networking.

• REFERENCES •

Agarwal, R., Tiwari, M., & Mukherjee, S. (2007). Artificial immune system based approach for solving resource constraint project scheduling problem. The International Journal of Advanced Manufacturing Technology, 34(5), 584-593.

Benedetto, H., Bernardes, M. M. S., & Vieira, D. (2016). Guidelines for the Deployment of Criteria for Selection of Strategic Projects in Design. The Journal of Modern Project Management, 4(2).

Beydoun, G., Low, G., Tran, N., & Bogg, P. (2011). Development of a peer-to-peer information sharing system using ontologies. Expert Systems with Applications, 38(8), 9352-9364.

Beydoun, G., Tran, N., Low, G., & Henderson-Sellers, B. (2006). Foundations of ontology-based MAS methodologies Agent-Oriented Information Systems III (pp. 111-123): Springer.

Bouleimen, K., & Lecocq, H. (2003). A new efficient simulated annealing algorithm for the resource-constrained project scheduling problem and its multiple mode version. European Journal of Operational Research, 149(2), 268-281.

Brucker, P., Knust, S., Schoo, A., & Thiele, O. (1998). A branch and bound algorithm for the resource-constrained project scheduling problem. European Journal of Operational Research, 107, 272-288.

Demeulemeester, E., & Herroelen, W. (1992). A branch-and-bound procedure for multiple resource-constrained project scheduling problem. Management Science, 38, 1803-1818.

Demeulemeester, E., & Herroelen, W. (1997). A new benchmark results for the resource-constrained project scheduling problem. Management Science 43, 1485-1492.

Engemann, K. (2014). Review of 'Measuring Data Quality for Ongoing Improvement: A Data Quality Assessment Framework'. Benchmarking: An International Journal, 21(3), 9-9.

Gurigallu, M. S. (2014). Software Architectural Pattern-Model View Controller to improve the Cost, Performance and Reliability of an Business Application. International Journal of Computer Science and Business Informatics, 10(1).

Gutierrez, O., & Friedman, D. H. (2005). Managing project expectations in human services information systems implementations: The case of homeless management information systems. International Journal of Project Management, 23(7), 513-523.

Hartmann, S., & Kolisch, R. (2000). Experimental evaluation of state-of-the-art heuristics for the resource-constrained project scheduling problem. European Journal of Operational Research(127), 394-407.

Hutchings, J. F. (2004). Project Scheduling Handbook. New York: Marcel Dekker Inc.

Jaafari, A., & Manivong, K. (1998). Towards a smart project management information system. International Journal of Project Management, 16(4), 249-265.

Jedrzejowicz, P., & Ratajczak-Ropel, E. (2007). Agent-Based Approach to Solving the Resource Constrained Project Scheduling Problem Adaptive and Natural Computing Algorithms (pp. 480-487).

Kerzner, H. R. (2013). Project management: a systems approach to planning, scheduling, and controlling: Wiley.

Kolisch, R., & Hartmann, S. (2006). Experimental investigation of heuristics for resource-constrained project scheduling: An update. European Journal of Operational Research, 174(1), 23-37.

Love, P. E. D., & Irani, Z. (2003). A project management quality cost information system for the construction industry. Information & Management, 40(7), 649-661.

Mahadevan, B. (2000). Business models for Internet-based e-commerce: An anatomy. California management review, 42(4), 55-69.

McReynolds, C., Lawrence, J. C., & Pujet, N. (2013). Network cost analysis: Google Patents.

Mori, M., & Tseng, C. (1997). A genetic algorithm for multi-mode resource constrained project scheduling problem. European Journal of Operational Research, 100, 134-141.

Nazareth, T., Verma, S., Bhattacharya, S., & Bagchi, A. (1999). The multiple resource constrained project scheduling problem: A breadth-first approach. European Journal of Operational Research, 112(2), 347-366.

Özdamar, L., & Ulusoy, G. (1995). A survey on the resource-constrained project scheduling problem. IIE Transactions, 27, 574-586.

Panagiotakopoulos, D. (1977). A CPM Time-Cost Computational Algorithm for Arbitrary Activity Cost Functions. INFOR, 15(2), 183-195.

Pardalos , P. M., & Pitsoulis, L. (2000). Nonlinear Assignment Problems: Algorithms and Applications (Vol. 7). Dordrecht: Kluwer Academic Publishers.

Pardalos , P. M., Rendl, F., & Wolkowicz, H. (1994). The Quadratic Assignment Problem: A Survey and Recent Developments. In P. M. Pardalos & H. Wolkowicz (Eds.), Quadratic Assignment and Related Problems: American Mathematical Society.

Pearl, J. (2000). Causality: Models, Reasoning, and Inference Cambridge: Cambridge University Press.

Raymond, L., & Bergeron, F. (2008). Project management information systems: An empirical study of their impact on project managers and project success. International Journal of Project Management, 26(2), 213-220.

Shimbo, M., & Ishida, T. (2003). Controlling the learning process of real-time heuristic search. Artificial Intelligence, 146(1), 1-41.

Sprecher, A. (2002). Network decomposition techniques for resource-constrained project scheduling. Operational Research Society, 53, 405-414.

Tan, H., Liu, X. L., Li, X. X., & Zhao, G. Q. (2014). Electronic Data Interchange on Logistics System Based on Embedded Linux. Applied Mechanics and Materials, 513, 2129-2132.

Tereso, A. P., Araujo, M. M. T., & Elmaghraby, S. E. (2004). Adaptive resource allocation in multimodal activity networks. International Journal of Production Economics, 92(1), 1-10.

Vanhoucke, M. (2013). Project baseline scheduling: An overview of past experiences. Journal of Modern Project Management, 1(2), 18-27.

Vanhoucke, M. (2014). Blended learning in project management: An overview of the operations research & scheduling group. Journal of Modern Project Management, 1(3), 108-121.

Vanhoucke, M., Coelho, J., & Batselier, J. (2016). An overview of project data for integrated project management and control. Journal of Modern Project Management, 3(2), 6-21.

Venkatesh, V., Brown, S. A., & Bala, H. (2013). Bridging the qualitative-quantitative divide: Guidelines for conducting mixed methods research in information systems. MIS Quarterly, 37(1), 21-54.

Wateridge, J. (1999). The role of configuration management in the development and management of Information Systems/Technology (IS/IT) projects. International Journal of Project Management, 17(4), 237-241.

Xu, D., Wijesooriya, C., Wang, Y.-G., & Beydoun, G. (2011). Outbound logistics exception monitoring: A multi-perspective ontologies' approach with intelligent agents. Expert Systems with Applications, 38(11), 13604-13611.

Zamani, R. (2010a). An Accelerating Two-Layer Anchor Search With Application to the Resource-Constrained Project Scheduling Problem. IEEE Transactions on Evolutionary Computation, 14(6), 975 - 984.

Zamani, R. (2010b). A parallel complete anytime procedure for project scheduling under multiple resource constraints International Journal of Advanced Manufacturing, 50(1-4), 353-362.

Zamani, R. (2011). A hybrid decomposition procedure for scheduling projects under multiple resource constraints. Operational Research, 11(1), 93-111.

Zamani, R. (2013a). A competitive magnet-based genetic algorithm for solving the resource-constrained project scheduling problem. European Journal of Operational Research, 229(2), 552-559.

Zamani, R. (2013b). An evolutionary search procedure for optimizing time–cost performance of projects under multiple renewable resource constraints. Computers & Industrial Engineering, 66(2), 451-460.

Zamani, R., & Shue, L. Y. (1998). Solving project scheduling problems with heuristic learning algorithm. Operational Research Society.