Towards Increasing Web Application Development Productivity through Object-Oriented Framework

Mohd Nuruzzaman, Azham Hussain, and Hatim Mohd Tahir

***Abstract*—Software development remains complex, expensive and risky. Thus, the need for reusability has become obvious. Through reusability web developers/engineers can save countless hours and millions of dollars in development cost. Object-oriented (OO) framework mechanism provides a new vehicle for reuse and enables web engineers to customize or reuse various aspects in web engineering such as customizing one or more elements of user interface, automatically generate event handlers, data handlers or related data elements based on changes made by them. The OO framework mechanism includes tags and scripts that predefine some generic web application activities and a common programming interface for generating a framework customization environment. This study provides a novel solution to produce high quality web applications within a shortest development timeframe through the means of customization, reusability, extensibility and flexibility. At the end, this study will conduct a comprehensive evaluation on the proposed OO framework. Building on previous works, this study emphasized the reuse of design, code and testing as a tool to uncover strengths and weaknesses of the OO framework for dynamic web engineering.**

***Index Terms*—Object-oriented framework, software reusability, software customization, web application design, component-based software engineering.**

I. Introduction

Object-oriented based web engineering is a relatively immature area of software engineering. Web application development is different from traditional software development as it focuses on visual creativity [1]. Object-Oriented Software Engineering (OOSE) including analysis, design, implementation, testing and documentation enables web engineers to repeat development phases and avoid potential failure in the fast changing landscape of the current ubiquitous web. This revolution in the software industry is making it easier for software engineers to write, distribute and extend software packages, and has also made a significant impact to software engineers working in industrial web applications.

In the past several years, many industrial web application software have been developed. Unfortunately, most of them were procedure-oriented, thus making them unsuitable for reuse and customization effectively as well as becoming more and more complicated. Considering this, efforts have been made to push legacy system software into the new OO technology development. There are many repeated works in this development, particularly in the design phase [2]. An approach is needed to achieve reusability, extensibility and reliability in web application development; only then web engineers/developers can reuse design as well as implementation. The need for software reuse has become evident because complex software remains difficult to implement, expensive to develop and risky to maintain. The idea behind reuse is not to develop anything that already exists, but just reuse it. This will lead to shortened development time, reduced complexity, increased productivity, extensibility and reliability of web applications [3]. Web engineers do not need to know how or when they have to call a function, the framework does it itself. The communication between components inside a framework is already defined and web engineers do not need to be concerned about it [4].

Framework-based software development significantly improves productivity, faster development and reduces complexity. Most of the projects fail because of poor design and quality of framework, developers and project management issues. Therefore, it is worthwhile to investigate current problems in application framework development and search for new solutions. Key factors that must be considered during OO application framework design and development include complexity of classes, complexity of object collaboration, difficulties in using framework and lack of clarity of requirements.

One of the most effective solutions to obtain reusable and extensible development is object-oriented application framework. Major goals of object-oriented technology (OOT) are abstraction, modularity and reusability of code by introducing standard interface, inheritance and component-based development. Class libraries provide well-defined and tested reusable components, but this mainly implies reuse of code and little reuse of analysis and design. To increase the potential for reuse, this study proposed object-oriented application framework as a web application development solution to overcome the above mentioned problems of dynamic web engineering.

II. Motivation

According to Hevner [5], currently web engineers face three intractable problems—domain/system complexity, increased development time and cost. It is agreed that complexity is an ever-present barrier in software development, customization, flexibility and reliability issues. This is due to the buildup of low-level details and the complex relationships among them that quickly exceed understanding by web engineers. Today's web engineers have no effective means to determine the full functional and non-functional behavior of program written by them or

Manuscript received September 4, 2012; revised November 18, 2012.

Mohd Nuruzzaman, Azham Hussain and Hatim Mohd Tahir are all with the School of Computing, College of Arts and Sciences, University Utara Malaysia (e-mail: nayan@egudang.com).

others. Furthermore, no testing process, no matter how systematic, can it validate the full behavior of program in all circumstances of use. The predictable result is that developed web applications are often fielded with unforeseen errors and vulnerabilities that no amount of trying harder can prevent. The problem is closely related to complexity which drives development time and cost, besides destroying modularity and reducing software quality [5]. As a result, web engineers, developers and programmers require a great deal of development efforts and time to develop today's web applications to satisfy customers, as well as to fulfill their requirements and delivery on-time. A new object-oriented web application framework is required which will enable more efficient, effective and faster development of reliable web applications with lower cost. Framework development by integrating object-oriented technology (OOT) and component-based development (CBD) for dynamic web engineering could achieve software reuse [6].

For the current literature in software reuse, customization has become evident. Research works have been done in the direction of developing extensible, reusable and customizable dynamic web application framework. Each of them is dealing with a different concern, operating levels of abstraction and degree of formality [7]. It is known that a universal application framework cannot exist due to various concerns, requirements and domains. Model-driven or visual object sharing techniques may be used to answer the need for supporting creation of flexible, extensible, customizable and stakeholder-oriented architecture. A big picture behind the approach to minimize dynamic web application complexity, development time and cost will be presented in this study.

III. Objectives

The objectives of this study are as follows:

- To identify the sources of software complexity in dynamic web engineering in order to achieve reusability, extensibility, modularity and customization of web applications within a shortest development timeframe. This is due to the fact that complexity is the biggest obstacle in software development.
- To design a new application framework to produce extensible software components that support GUI-based dynamic web application framework.
- To make recommendation based on the evaluation of the proposed OO dynamic web application framework through software metrics and case study in the area of supply chain and logistics management.

This study does not judge whether the proposed object-oriented framework is appropriate or not in a financial perspective. However, this study will provide a complete development process for web engineers to develop dynamic web applications. This study will also specify how to reuse, customize web applications through the framework implementation.

IV. Related Works

A software framework is a collection of libraries providing a defined API. It is a universal, reusable software platform used to develop applications, products and solutions. The aim of software framework is to facilitate the development process and make easy of software development by allowing software engineers to devote their time to meeting requirements rather than dealing with low-level details. For example, to develop a warehouse management system, web engineers/programmers can focus on the location utilization or distribution process rather than the mechanism of request handling or state management.

In an object-oriented software environment, a framework consists of abstract and concrete classes. It also consists of composing and sub-classing the existing classes [8]. According to Wallace [9], a software framework relies on the Hollywood Principle: "*Don't call us, we'll call you*". This means that user-defined classes inherit from predefined classes in the framework. Web engineers handle this by implementing superclass abstract methods in a component. By doing so, developers spend less time coding, less debugging and spend more time on value-added development and concentrate on the business specific problems.

According to Carlos and Pedro [3], in OOSE—"*an OO framework is a reusable semi-complete application, a set of objects that collaborate to carry out a set of responsibilities for an application system"*. The OO framework provides highly effective mechanism (reusability) by utilizing the domain knowledge and effort of experienced software engineers in order to avoid recreating common solutions to current application requirements [10]. Like design patterns, an application framework emphasizes working with a certain domain. While design patterns are smaller and more general units, a framework may use one or more design patterns, but the reverse is never true [20]. Design patterns are descriptions of collaboration between objects and classes that are customized to solve a general design problem in a particular context [11]. The design pattern Model-View-Controller (MVC) is widely used for developing web applications. Design patterns and frameworks help to improve software quality, performance and reduce development time [12].

Fayad et al., [23] mentioned that design patterns are conceptual models that capture core knowledge and concepts of the problems. Knowledge and concepts are presented in abstraction levels that make them reusable whenever the problem occurs. In this paper, they proposed a new approach to use analysis and design patterns to develop a software system. Their approach transforms analysis patterns into a design model by composing the appropriate set of design patterns. The approach can be generalized to transform a collection of analysis patterns into several design components. Then each design component can be developed separately using existing design patterns.

Lee, Thin and Liu [6] introduced problematic issues in software development by conducting a study on Java-based software framework on manufacturing domain system. The research aimed to provide further evidence on how an object-oriented application framework (OOAF) can be used to assist software development to enable large-scale software reuse to reduce cost and time of developing manufacturing domain systems. The research focused on production

management and statistical quality control, two subdomains of the manufacturing domain. In addition, it also presented design pattern as defined by Fayad [13] to implement application framework based on the template method. The idea is to define a template method in a superclass that defines the skeleton of an algorithm, with its varying and unvarying parts. To the extent that this research is exploratory, they provided a unique solution for developing an OO application framework on the manufacturing domain by integration of OOT.

Riehle [15] proposed role modeling for framework design. Role modeling for framework design is an evolutionary extension of class-based modeling framework. This approach enhances class-based modeling with role modeling concepts. In this approach, an object plays roles that are described by role types. An object can play multiple roles, so that the class of that object has several role types. Moreover, objects collaborate for several different purposes called object collaboration tasks; each is described by a role model. "A class model is composed all of relevant role models to describe how instances of its classes collaborate. Describing classes as compositions of role types and class models as compositions of role models reduces class and object collaboration complexity"- as Riehle [15] described. Role types are key to defining the requirements of a framework. A role model provides role types that a framework has to play to make proper use of it.

Laakso and Niemi [16] presented how to evaluate AJAX enabled Java based dynamic web application framework and devise an economical method to conduct scenario-based evaluation. The evaluation was conducted from a limited number of candidates. Their opinion proved that the results of the method they proposed also could be applicable to other projects. Another review proposed a three-step process to determine whether or not a framework is appropriate, and look at the limitations, hooks and the amount of uncertainty in using a framework. Framework evaluation could also be validated with the help of case studies. Riehle [15] presented three case studies that show how framework design works in practice. Each of the case studies validated from a different angle and compared with a traditional class-based framework design.

V. Research Approach

The primary purpose of this study is to propose a novel framework to reduce development time and cost, besides, increasing component reusability, extensibility, modularity and flexibility. In order to do so, it is necessary to identify the processes and evaluation criteria that support the framework. In our approach, there are three main phases such as—theoretical framework, design & development and experimentation & evaluation. This approach is suitable to be applied in developing an OO framework model, where theories or concepts were derived from theoretical findings.

A. Theoretical Framework

Given the broad scope of software framework, a comprehensive study was required to take into account many different aspects of software framework and architecture. This consists of identifying past and current trends and considered the features of software toolkits. Literature review, ideas, issues and problems related to object-oriented framework were collected from books, proceeding, journals, articles, white papers, reports and the Internet. These provided a good foundation of knowledge to understand the most important requirements, definition of the requirements, brief history of the development tools and environments, strength and weaknesses of the framework reuse solutions.

This phase includes two studies—theoretical study and empirical study. The theoretical study focused on three main issues: previous OO frameworks, processes and techniques to design reusable, customizable, extensible and flexible an OO framework to solve the current problems in web application development. The empirical study focused on usability and case study to investigate these elements in the real-life environment and evaluate the framework reliability and performance.

We identified and analyzed the common processes, concepts of software design and evaluation criteria that would be used in the next phases. Additionally, deep analysis on the existing OO frameworks has been carried out. Also the issues related to framework design strategies, reusability and customization mechanisms, guidance and templates that facilitate implementing an OO framework has been investigated. We found that the established OO methodologies do not provide an adequate framework to support design reuse. Besides, they do not provide developers guidelines for navigational and user interface design for dynamic web application development. Therefore, it is important to understand the current practices and problems that are faced by web engineers. This study aims to undertake these problems by introducing a novel mechanism which establishes domain analysis, user interface design reuse, extensibility, flexibility and quality of software products.

B. Design and Development

Creating an effective web application requires some degree of design. To design an application, it involves well understanding of framework process, complexity, techniques and a suitable solution to the problem. It is very difficult to understand the concepts of a software component by reading its specification. Thus, web engineers can only utilize the framework partially and much effort required understanding the given OO framework limitation. To gain full understanding of an object in the component, web engineers must see the object static context, behavior and knowledge of the component. To support this perspective, there are three phases where reusability could be applied—analysis phase, design phase and development phase.

C. Analysis Phase

The analysis phase aims to capture the requirements imposed on the software framework as well as modeling the application. Introducing reuse at the analysis phase has a great impact on software development process. All requirements have to be formulated as general as possible, because too detailed requirements do not map with existing requirements. Analyzing requirements/ features is a good

way to identify potential areas for reuse. It involves understanding of the studied domain, technical knowledge, creation of various alternatives and evaluation of proposed alternative solutions. To increase the suitability for reuse, customization and modularity, this study identified sources of software complexities and described necessary functionalities and performance.

D. Design Phase

Our proposed framework design is modularization based which provides flexibility and comprehensibility in a shortest timeframe. This study divided the design phase into two sub phases—architectural design and framework design. The architectural design aims to define a high-level strategy for solving problems and provides a road map for implementing the solution. In this phase, a unique architectural design has been developed for extensibility, compatibility and flexibility of web application development as shown in Fig. 1(a). System components, functionalities, structural relationships and dynamic interaction among components have been identified to achieve the objectives of this phase. Requirements defined are measurable and can be validated at the evaluation phase. All requirements have been captured from a specific domain (supply chain and logistics management) and its technical environment. New functionalities and innovative user interface features were also proposed in the proposed dynamic web application framework. The framework design consists of two parts—Preliminary design and detailed design. The preliminary design will initialize the generation and exploration of the OO framework design ideas. Detailed design will identify the problems and solutions of the given problems. Therefore, it is important to know what are the solutions already exist and identify the solution suitable to the problem. The detailed design will show the central functionality of the framework. Fig 1(b) shows the preliminary prototype design of the OO application framework.

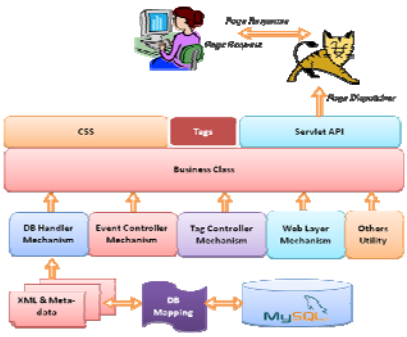

Fig. 1. Architectural design of the framework.

Design patterns will be used to solve design problems and blueprint for the implementation of the proposed dynamic and interactive web application framework. Design of UI, object collaboration and visual object sharing mechanism will be determined to be reused to produce the desire behavior or robust structure. In this stage, our effort is directed towards visual object sharing mechanism which generate codes, event handling, generic web application scripts so that web engineers/developers can concentrate on the business logic and value-added services.

E. Development Phase

Development of new software components that adapt the OO framework will follow the architectural strategy of the framework as well as adding new library components to the framework. All components in the OO framework will be designed and implemented within the same standards and strategy. Reusing an OO framework is partially done by component reuse that is already defined in the framework. When performing adaptation, new objects and classes may be identified. This may be introduced in detailed design phase.

The real-time Java programming language will be used to develop the proposed OO application framework for dynamic web engineering. “Spring Tool Suite” as Java IDE, Tomcat 6.2 as web application server and JDK 6.20 as Java complier will be used in the development environment. We concentrate on using an OO language because it has the necessary features for developing an OO framework and most of the industrial products are written in OO style. Developing system will be used to demonstrate the feasibility of the design and usability of the functionalities. The design and development phase will be repeated until acceptance test scores are satisfied.

VI. Experimentation and Evaluation

Once the framework is built, experiment and evaluation phase begins. The aim of this phase is to evaluate the effectiveness, reliability and acceptability of the proposed framework in the real-life environment. Experiments in software reuse include studies of indexing methods for reusable components and correlational studies of the relationship between reuse, quality and productivity. There are various methods of evaluating reusability of framework components. We proposed a method of quantifying reusability of software components by calculating various matrix values, normalizing them and adding together the resulting normalized values [17]. On the other hand, to evaluate software components which were programmed with non-disclosed source codes, only that interface elements of those components will be used. In addition, reusability will be evaluated based on the similarity between the rankings of each evaluated value. If multiple components are reused, the corresponding result from a web engineer subjectively evaluates the reusability [18].

Knowledge gained from the development process can be consolidated into a case study that describes the rationale, process and experiences [19]. The evaluation process will also be done through a case study and let web developer/engineers to develop some modules and observe the usability, performance on individuals. For evaluation, this study will focus on both professional and novice web engineers who will be asked to create a high-level design and develop for a web application module through the proposed OO framework.

VII. Productivity Development Strategy

The lower level holds mechanisms for the OO framework development strategy. In this study, we introduced a novel

reusability mechanism as a key requirement to increase web application development productivity. In our mechanism, we introduced visual object sharing technique which plays a key role and enables reusability, scalability, flexibility and customization.

The scalability issue arises when the web application framework becomes large and complex. Working with such framework is very difficult due to too many classes, large size classes, complexity of object collaboration and complexity of relationship among them. Hierarchical representation is one of the most effective techniques. This technique enables a program to be divided into multiple modules, so that class size will become small. In addition, Flexibility allows web engineers to reduce the complexity of the classes, simplifying the objects collaboration and increase reusability. By integrating this technique with hierarchical representation, a scalable, extensible, flexible and reusable framework can be developed.

Object-oriented languages improve reusability in number of ways, such as inheritance, polymorphism, delegation and framework. However, there have been few attempts to improve the reusability through visual objects. Therefore, our aim is to develop a technique to enhance reusability via visual object sharing which will also allow specific code generation. This will contribute to the improvement of the web application development productivity.

In our proposed mechanism, visual object sharing technique will allow a number of modules to share one object instance as their component by changing the relationship and property defined between them. Without visual object sharing, each module has own independent component. With visual object sharing, multiple modules can share only one object instance. “Visual objects sharing” is based on iconic objects or GUI window objects to construct web application user interface modules.

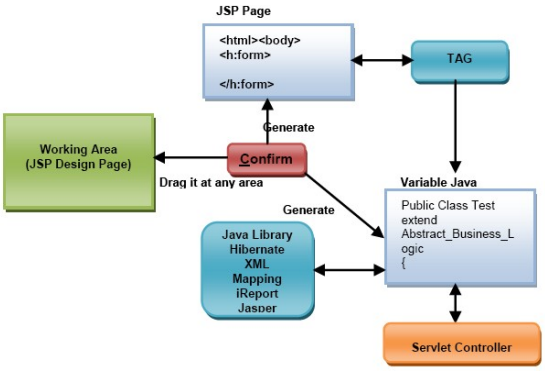

Fig. 2. Reusability mechanism through visual object sharing.

To create a GUI module web engineers should drag and drop objects into GUI window only. Each object has its own properties which contain visual object ID, some public methods and sub-components. Web engineers should be able to customize the object properties or define new object behavior. These apply in the same way to other objects. JSP page, business class, controller class, event handler, public and abstract methods are created when the module is executed. Fig. 2 shows a sample visual object mechanism on how it collaborates with other classes and objects. Each module abstracts its components. Therefore, when a module instance is created, instances of components are created, and when a module instance is destroyed, component instances are destroyed. Each module also has its own chain of control and could be invoked from outside of the module. To share the same visual object with the same behavior into other web modules, web engineers just need to specify the object ID into visual object property window. Invalid object ID call will lead to error during run-time execution.

VIII. Discussion

The complexity problem of classes is resolved by hierarchical representation technique which divides the large and complex classes into smaller and multiple modules. However, this would lead to increasing number of classes as a result of the division of large classes. Therefore, visual object sharing technique was used to minimize this problem. By object sharing, a set of classes could have separation of instance object collaboration, which cannot be provided by a single class. Object sharing mechanism defines precisely how an object of a class interacts with other classes in a given context or shares the same behavior by changing relationship and property defined between them. Thus, our visual object sharing mechanism could provide a great flexibility and as one of the key techniques to reduce the software crisis.

Currently, the predicted level of reuse has not yet been reached. This study will be implemented and conduct a comprehensive evaluation on the proposed OO framework for developing dynamic web applications. Building on previous works, this study emphasized the reuse of design, code and testing as a tool to uncover strengths and weaknesses of the OO framework.

IX. Conclusion

An OO framework promises reusability, higher productivity and shorter time-to-market for developing OO-based applications. However, many projects show that these are not always easy to be met. This study addresses relevant problems, several key problems on why framework promises fail, describes framework concepts and discusses related works. Some areas such as service-based resource composition, low level activity identification and detailed design needed to be explored more. Future work should be aimed to check capability and navigation of modules combination on top of the proposed OO framework. During reusing, some object conflicts may occur and some calling invalid object ID may generate error during execution. Therefore, it is necessary to show web engineers what kind of conflicts/ errors occur. To overcome this problem, searching object ID may be applied.

References

[1] S. Murugesan, Y. Deshpande, S. Hanse, and A. Ginige, “Web engineering: Introduction and perspectives,” in GINIGE, Ed., *Web Engineering*, Idea Group, 2005.

[2] S. Pandit, S. Soman, and S. A. Khaparde, “Object-oriented design for power system applications,” *IEEE Computer Applications in Power*, vol. 13, no. 4, 2000.

[3] J. Carlos and A. Pedro, “Domain analysis of object-oriented frameworks in FrameDoc,” SEKE’02, Ischia, Italy, *Journal of ACM*, pp. 27-33, 2002.

[4] N. Laga, E. Bertin, and N. Crespi, "A web-based framework for rapid integration of enterprise applications," *Journal of ACM*, pp. 189-197, 2009.
[5] A. R. Hevner, R. C. Linger, R. W. Collins, and S. T. Prowell, "Next Generation Software Engineering," *Software Engineering Institute* (SEI), Carnegie Mellon, Pittsurgh, PA, CMU/SEI-2005-TR-015.
[6] S. P. Lee, S. K. Thin, and H. S. Liu, "Object-Oriented Application framework on manufacturing domain," *Malaysian Journal of Computer Science*, vol. 13, no. 1, pp. 56-64, 2000.
[7] I. Malavolta, "Providing support for creating next generation software architecture languages," *International Conference of Software Engineering* (ICSE' 10), *Journal of ACM*, Cape Town, South Africa, 2010, pp. 517-518.
[8] I. J. Hong and J. A. Landay, "An infrastructure approach to context-aware computing, in human-computer interaction," vol. 16. 2001.
[9] Wallace and Bruce, "A hole for every Component and every Component in its Hole," Existential Programming, 2011.
[10] R. E. Johonson and B. Foote, "Designing resuable classes," *Journal of Object-Oriented Programming*, vol. 1, no. 5, pp. 22-35, 1998.
[11] A. Stoev and A. Dimov, "Architectural framework for dynamic web-applications," *International Conference on Computer Systems and Technologies- CompSysTech'08*, vol. II. 10, pp. 1-6. 2008.
[12] C. Oscar and L. Angel, "The ODESew 2.0 semantic web application framework," *Journal of ACM*, pp. 1049-1050, Scotland, 2006.
[13] M. E. Fayad, D. C. Schmidt, R. E., and Johnson, "Building application frameworks: Object-oriented foundations of framework design-1999", John Wiley & Sons, Inc. New York, NY, USA.
[14] D. C. Schmidt, "Applying design patterns and frameworks to develop object-oriented communications software," in Handbook of Programming Languages, P. Salus, Ed. Macmillan Publishing Co., Inc., Indianapolis, vol. 1, 1997
[15] D. Riehle and G. Thomas, "Role model based framework design and integration," in *Proc of the 2000 Conference on object-oriented programming systems, languages and applications (OOPSLA)*, ACM Press, pp.117-133, 2000.
[16] T. Laakso and J. Niemi, "An evaluation of AJAX-enable java-based web application frameworks," Prec of MoMM 2008, Linz, Austria, pp. 431-437.
[17] R. Bandi, V. Vaishnavi, and D. Turk, "Predicting maintenance Performance using object-oriented design complexity metrics," *IEEE Transactions on Software Engineering*, vol. 29, no. 1, pp. 77-78, January, 2003.
[18] H. Yamamoto, H. Washizaki, and Y. Fukazawa, "A metrics suite for measuring resuabiluty of software components," Matsushita Electric Industrial Co., Ltd., Waseda University, Osaka, Japan, 2004.
[19] F. J. Nunamaker, M. Chen, and T. Purdin, "Systems development in information systems research," *Journal of Management Information Systems, Proc of the Twenty-Third Haiwaii International Conference on System Sciences*, vol. 7, no.3, pp. 89-106, 2001.
[20] E. Gamma, R. Helm, R. Johnson, and J. Vlissides, "Design patterns: Elements of reusable object-oriented software," Addison-Wesley Pub Co., 1997, MA.
[21] D. Schwabe, L. Esmeraldo, G. Rossi, and F. Lyardet, "Engineering web applications for reuse," *IEEE Multimedia*, vol. 8, no. 1, pp. 20-31, 2001
[22] M. H. Hazem, E. H. Wassim, M. Dana, D. Marwan, and F. Faysal, "An extensible software framework for building vehicle to vehicle applications," IWCMC'2010, ACM Press.
[23] M. E. Fayad, S. H. Hamza, and Y. Chen, "A framework for developing design models with analysis and design patterns," *Communication of IEEE*, 2005.

Mohammad Nuruzzaman obtained his Bachelor degree from University Utara Malaysia in 2010. He is majoring in software engineering. He was hired by Softworld Technology Sdn Bhd as a software engineer in May, 2009. His experience sparked a lifetime passion for learning, and he intends to carry on cutting-edge research and as practitioner in the software industry. Currently, he is working as a system analyst at ENT broadband. In the same time, Mr. Nuruzzaman is pursuing Master of Science at School of Computing, University Utara Malaysia. His research interests include requirements engineering, software visualization, modeling, simulation and mobile network security. He has several publication and award related to these areas.

Azham Bin Hussain obtained his PhD in computer science from University of Salford, UK in 2010. He was appointed as a programmer at Felda Prodata System Sdn Bhd. Currently, he is a senior lecturer at School of Computing, University Utara Malaysia. Mr. Azham received quite number of research grant. His research interests are in requirements engineering, project management, system design, development and extreme programming. He has several publication and award related to these areas.

Hatim Mohd Tahir obtained his Bachelor and Master of Computer science degree from Northrop University California, USA. Upon completion his diploma in computer science he was appointed as a programmer at FAMA in 1983. Since 1990, he becomes an associate professor at University Utara Malaysia. Mr. Hatim was project leader, member and consultant for several research projects. His current research on Mobile Ad-Hoc Network has funded by Ministry of Science Technology and Innovation (MOSTI). His research interests are information security, Intrusion detection, Mobile ad-hoc network and network security. He has several publication and award related to these areas.