Agent-Oriented Requirement Engineering for Mobile Application Development

https://doi.org/10.3991/ijim.v11i6.6760

WaiShiang Cheah(✉), Aida Shafreena bt Ahmad Puad, Puah ChinHong
Universiti Malaysia Sarawak (UNIMAS), Kota Samarahan, Sarawak, Malaysia
wscheah@unimas.my

Alfian Abdul Halim
Universiti Putra Malaysia, Serdang, Selangor, Malaysia

Abstract—Mobile application development is receiving much attention nowadays. With the enhancement of mobile application tools like an Android studio, etc. and kinds of online support, the development of the mobile application is getting easier. Indeed, mobile application development is not a trivial task. When given a particular problem, a novice mobile programmer will commonly sketch the mobile interface followed by coding. The rapid prototyping technique and trial from errors have led to issues such as poor domain understanding. We argue that a complete understanding of the domain is needed for mobile application development. Hence, requirements engineering is an important phase. This paper introduces a technique to assist mobile application development through Agent-Oriented Requirements Engineering (AORE). AORE consists of goal modelling to analyse and understand a mobile-based project. With goal modelling, AORE allows a modeller to identify and analyse the functionalities and non-functionalities of the system and present a holistic view of the proposed system. It showcases the services, operations and constraints of the proposed system. AORE is a useful part of the development phase and can complement current steps in mobile application development lifecycle.

Keywords—requirement engineering for mobile application, agent-oriented modelling, functional and non-functional requirement

1 Introduction

The emerging of mobile technology like Android and ios has changed the way people use and deal with technology. With the mobile application, it changes the way people interact with the digital information; improve social interaction; faster transform people in digital society. In this case, people use a mobile phone to access to news; attend the meeting; attend course online; entertainment and interact with others. Although lots of mobile application tools have been introduced, mobile application development is still suffering from some inadequacies. This is due to the nature that, the evolution of mobile technology is fast with the emerging of various standards, protocol, network technology and mobile platforms has been introduced [1]. Hence a traditional software development method is unsuitable to cope with those changes [2]. In addition, the mobile application development needs to deal with business-related challenges, conflicting customer interest, innovation issues, integration with the existing systems. As a result, there is a need to introduce a new mechanism or methods in order to support the complex system development for mobile application. Inline with the works to explore agent models for software application [12,13,14,15,16], this paper introduce agent-oriented requirement engineering for mobile application development.

Requirements engineering (RE) is the first phase in software development where it is the process of understanding user requirements and needs. Requirement engineering is defined as a process where one or more stakeholder needs and their environment are determined to solve a specific problem [10]. This is a crucial step as failure to effectively execute this step can result in project failure. Meanwhile, good requirements definition on the other hand leads to a high-quality solution and reduces the cost of changes [10].

The agent-oriented requirement engineering consists of several steps and support by goal modelling, role model and organisation model to analyse and understand mobile-based projects. With the goal modelling, a modeller is able to identify and analyse the functional as well as non- functional aspects of the system, able to present a holistic view of the system, as well as able to showcase the services, operations and constraints of the system. In addition, it can also serve as a contract between stakeholders.

The following section presents our proposed agent-oriented requirement engineering (AORE). The proposed AORE is validated in Section three of this paper through a walkthrough example in understanding and building the Kuching FloraLover mobile application. In addition, AOSE is then compared with other methods in engineering mobile application in Section four, where AORE's advantages are highlighted within the mobile application development lifecycle. The section four analyse the proposed AORE with existing methods and show how the AORE can serve as the first phase in mobile application development. Also, it presents how AORE can be further extended to form a complete methodology for mobile application development. Section five presents related works. The paper is concluded in Section six.

2 Agent-oriented Requirement Engineering

According to [10], a typical process for capturing the requirements or requirement engineering consists of several steps. They are elicitation, analysis, specification and validation. The elicitation step determines the user's and customer's needs. The analysis step involves processes to understand the user requirements and modelling the desired behaviour. The specification step documents the behaviour of the proposed software system and the validation step checks the user specification. Each process is supported by a dedicated tool and method. For example, UML modelling is adopted at the analysis step whereas prototypes, walkthrough, interviews and simulation are tools used for validation.

Figure 1 presents the proposed requirement engineering process for mobile application development. It consists of the Process and the supported tool and techniques. The process for agent-oriented requirement engineering includes a subset of the steps namely elicitation, analysis, specification and validation. The process occurs iteratively (i.e. “agile way”) where developers can cross check between steps. The iterative or agile mechanism is claimed to be one of the most effective ways for mobile application development [10].

During the elicitation step, interviews can be conducted with the user to identify whether requirements have been met. In order to obtain the full and pertinent details during the interview sessions, users should be clear on what they want and are able to elaborate on the system functionalities. This however can be an issue if the technology is new or users are unfamiliar with the domain. Therefore, developers might find it beneficial to review similar systems in order to identify user needs. Upon identifying user needs, the analysis is conducted to understand and model the behaviour of the system.

The analysis step is done through goal modelling, role modelling and organisation model. Modelling system behaviour is part of the analysis and specification processes in AORE. During system behaviour modelling, the developer will firstly model a higher-level goal using the ROADMAP goal model. This is followed by the distribution of the actors, modelled through an organisation model. The organisation model presents interactions among cooperative players/ people who try to solve a problem. Then, a details goal model is presented through the i* goal modelling. This captures the detailed functionalities of the system, goal dependency and non-functionalities of the system. With goal modelling, user satisfaction can be analysed.

The ROADMAP and i* goal models are adopted in this work due to the following design principles: Firstly, it is important to present the requirements at a higher-level of abstraction where it can be understood by both technical and non-technical persons. Secondly, the model must be able to represent functional requirements of the system. Thirdly, the model must be able to represent non-functional requirements of the system. Finally, the model must be able to be mapped into a Design model. The goal modelling is introduced to present a higher level of abstraction of a system [12].

In AORE, a goal represents a system’s purpose. Achieving a goal involves fulfilling sub-goal, and the goal is served by an individual or a group of actors. An actor is a person who plays a specific role in an organisation. The actor has responsibilities and is responsible for serving the entire goal or sub-goals. The Goal model hence describes the purpose of the system (e.g. system functionality) at a higher-level of abstraction. Since goals can be decomposed into sub-goals, the goal of text extraction can be subdivided into the sub-goals of parser setting, generating a training set, parsing, validating, and handling local storage. In addition to functional goals, there are quality goals that represent non-functional requirements of the system. Achieving a goal consumes resources, and a goal is related to a particular role which indicates the actor or agent that is involved in achieving the goal [11].

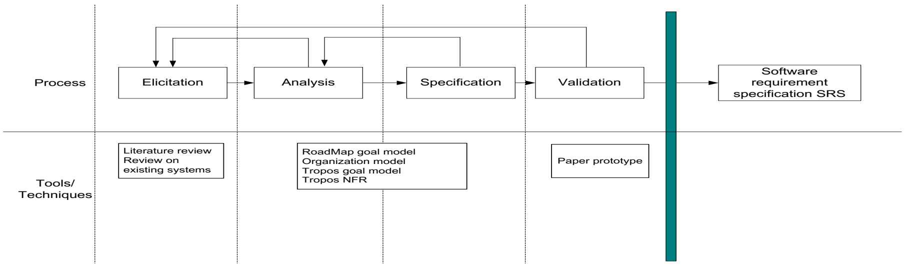


Fig. 1. Processes for agent-oriented requirement engineering [adopted from[10]]

Figure 2 shows the notation of the ROADMAP and i* goal models. The ROADMAP goal model will model a system with a goal(s), sub-goals, role and quality goal(s) and the relationship between goal and role. On the other hand, the i* goal model models a system with a goal(s), sub-goal(s), goal dependency, actor, plan and resources. The goal and sub-goals present the functionalities of a system. Meanwhile, the quality goal presents the non-functionality of a system like emotion, user satisfaction, privacy, security and so on.

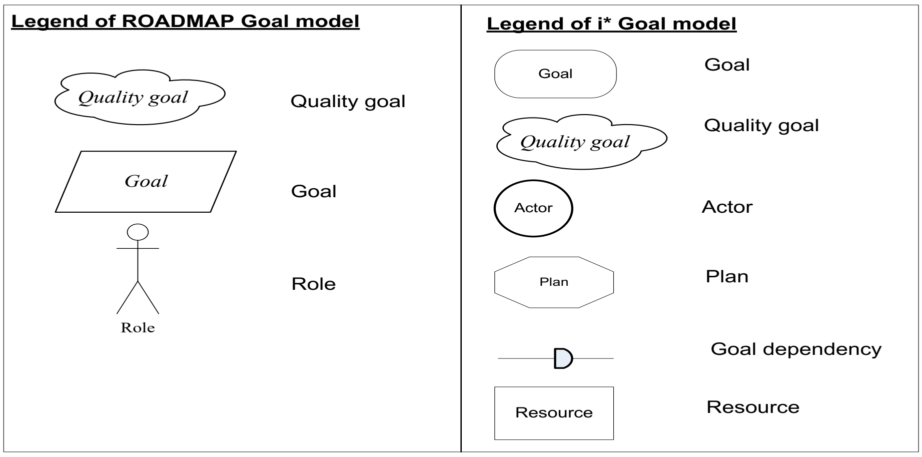


Fig. 2. Notation on goal models

Upon the modelling, we can validate through paper prototyping. In paper prototypes, the developer will design the system's user interface (UI) for obtaining user input and showing system output. The input presents the incoming messages the system will receive, whereas the output presents the outgoing messages from the system after processing. From the paper prototyping, the users can preview on the proposed system. The users can easily highlight the satisfaction of the system and comments on it.

A proposed AORE for mobile application is presented in this section. A walkthrough example of the proposed process is presented in the following sections.

The walkthrough example will validate the proposed agent oriented requirement engineering for mobile application development.

3 Case Study

This section presents a walkthrough case study of Kuching FloraLover, a mobile eco-tourism flora information system for the town of Kuching, in Sarawak, Malaysia. Kuching FloraLover is a mobile application that promotes the biodiversity tourism in Sarawak. It records flora information and the location of flora visitation sites (e.g. Event, flora activities, flora package Park, garden etc) in Kuching. It is designed to attract plant lovers so they can enjoy the unique types of plantation in Sarawak. In line with the effort to promote the Dayak tourism industry, the proposed system will relate the location of the flora visit with Dayak homestay and restaurants.

With Kuching FloraLover, visitors are able to perform the following tasks. The following tasks are system functionalities based on the review of similar existing systems: 1) View on flora information, comments on the flora, locality of the flora types, events, news on flora; 2) View of tourism facilities like place, transport and local food; 3) Preview the gallery on the flora type, galley on the places for flora; 4) Preview of other flora lover; 5) Capture of the flora enthusiasts and share with others; Working on to do list; 6) Search for the locality on the flora type; 7) View of promotion items; Share the information to others and 8) Ask for help

4 Walkthrough Example

This section presents the validation of the proposed AORE, which is similar to the validation done through case studies such as in [3, 5, 6]. In order to identify the requirements of the system, first, we review similar existing mobile tourism applications and systems. From the review, the functionalities of the systems are identified as listed in the previous section. Then, modelling is conducted. Finally, the paper prototype sketch is produced. The paper prototype consists of the system’s user interface (UI) design and the system flow.

Figure 3 shows the ROADMAP goal model for Kuching FloraLover. It presents an overall purpose of the system. The goals are derived based on the elicitation results as shown in the previous section. In general, the Kuching FloraLover is a mobile application that is developed to serve the purpose of 'handle flora lover'. In order to achieve the goal of 'handle flora lover', several actors are involved. They are tourism, tourism provider, flora system provider and flora experts. Six main sub-goals are required for this system. They are 'inform general information', 'inform promotion', 'provide flora information', 'handle ask for help', 'provide flora tool' and 'present who lovers'. The goal of 'inform general information' is decomposed into the sub-goals of 'inform comments', 'inform news' and 'inform events'. The goal of 'provide flora information' is decomposed into 'describe flora type', 'present photo' and 'present locality'. In addition, the sub-goal of 'present locality' is further decomposed into 'inform details place', 'inform stay', 'inform food' and 'inform transport'. The goal of 'provide flora tool' is decomposed into the sub-goals of 'handle to do list' and 'capture flora lover'. Finally, two main quality goals are modelled in Figure 3. They are usability and sustainability.

Figure 4 shows the organisation model for Kuching FloraLover. There are two main actors namely Tourism and Flora System Provider. The tourism interacts with the flora system provider on flora services. Flora System Provider further interacts with the other actors such as Flora Expert, News Provider, Marketing Officer etc., in order to fulfil service requests by Tourism. In other words, these interactions by Flora System Provider serve the main goal of 'Handle Flora Lover'. For details regarding the actors' responsibilities, please refer to Table 1. Table 1 presents a simplified version of the role model to model the responsibility of actor for a system.

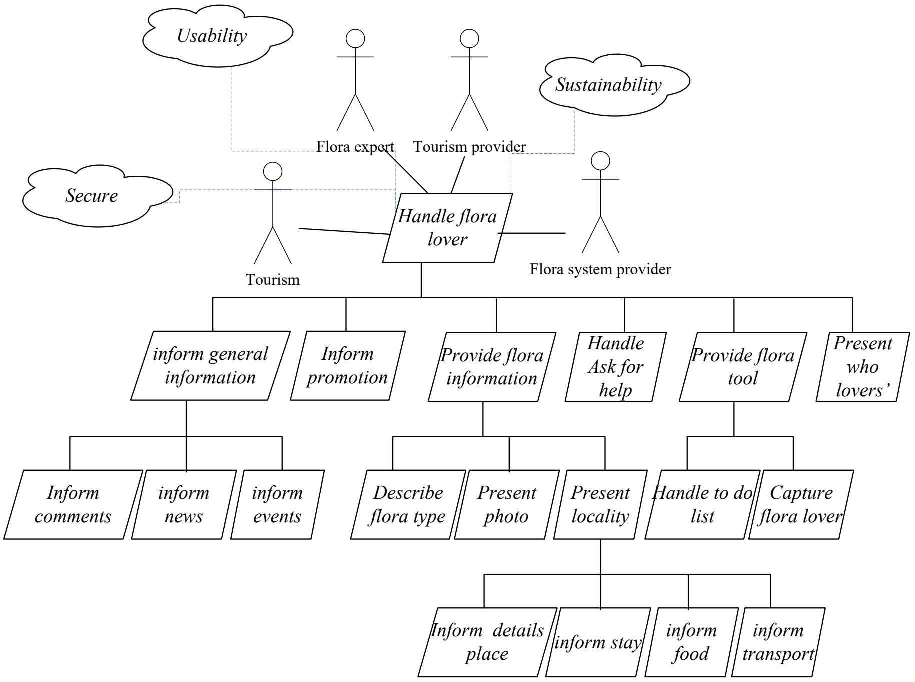


Fig. 3. ROADMAP goal model for Kuching FloraLover

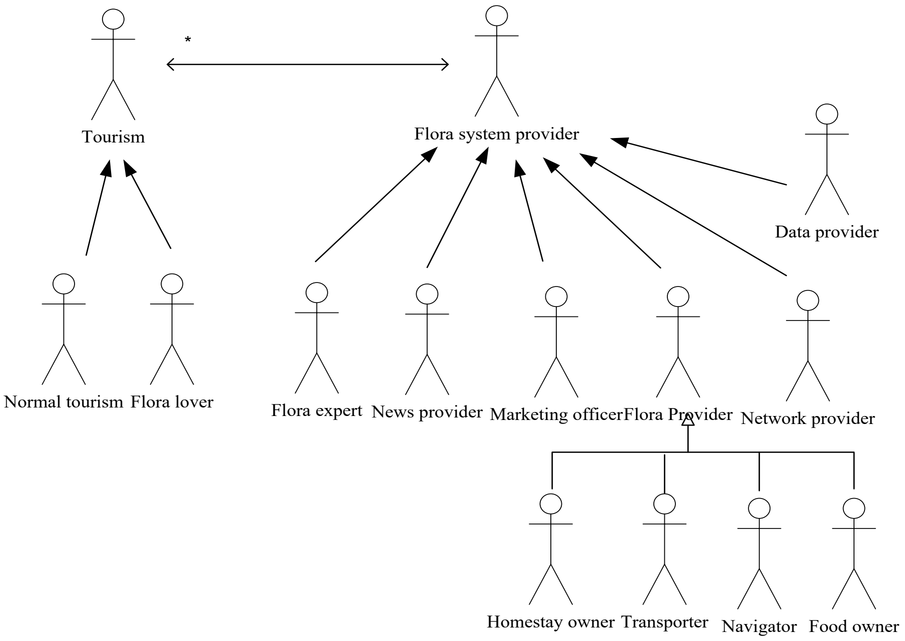


Fig. 4. Organisation model for Kuching FloraLover

Table 1. Role with Responsibility for Kuching FloraLover

Tourism:	A person who interacts with Flora System Provider for flora services such as knowing the flora information, events, comments, etc. There are two types of Tourism actors: (1) 'Normal Tourism' who interacts with Flora System Provider for travel information, and (2) A Flora Lover' who has specific interest and knowledge about flora.
Flora System Provider:	A system provider for flora system. The provider receives a request from Tourism and serves Tourism.
Flora expert:	A person who provides flora information and scientific knowledge.
News provider:	A person that is responsible for updating the flora news, and comments.
Marketing officer:	A person is responsible for collecting promotional details from flora provider.
Flora provider:	A person is responsible for handling the flora provider, information, locality and events. The person who interacts with homestay owner, transporter, tour guide, food owner.
Homestay owner:	A person who is responsible for providing accommodation information to Tourism.
Transporter:	A person who is responsible for providing transport information to Flora provider.
Navigator/Tour guide:	A person is responsible for navigation tasks for Tourism during flora visits.
Food owner:	A person is responsible for serving food and food related information to Tourism.
Network provider:	A person who is responsible for handling the network services for the system.
Data provider:	A person who is responsible for keeping and managing the record on flora information.

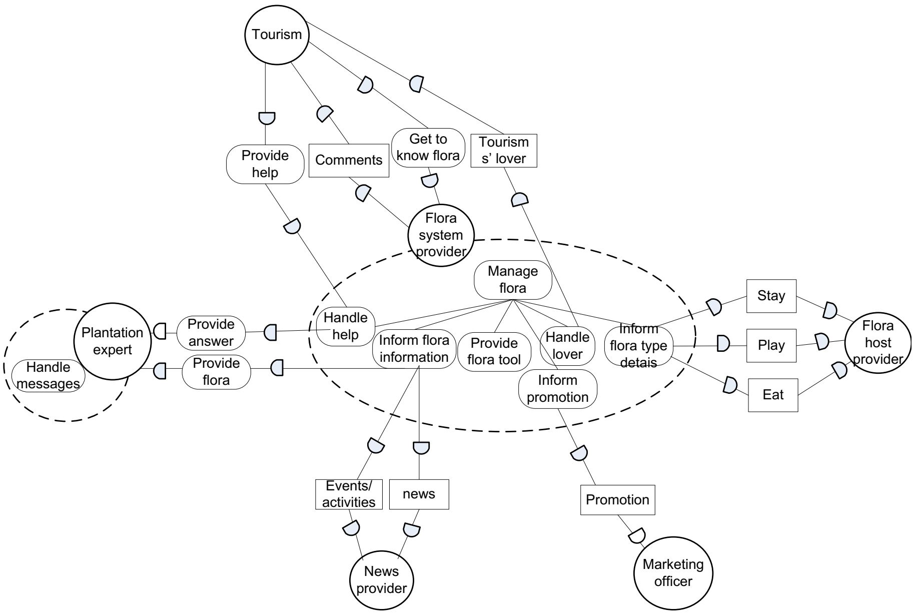


Fig. 5. i* goal model for Kuching FloraLover

From the overview goal model of Kuching FloraLover (Figure 3), a detailed goal model is presented (Figure 5). Figure 5 shows the goal dependency based on the i* goal model for Kuching FloraLover. It consists of interactions between the Flora System Provider and the roles such as 'plantation expert', 'flora host provider', 'tourism', 'news provider' and 'marketing officer'. Tourism will depend on 'flora system provider' to achieve a goal of 'provide help' and 'get to know flora' where as the 'flora system provider' will depend on ‘tourism' on comments and 'tourism lover'. The 'flora system provider' depends on 'flora host provider' on stay, play and eat information. This will be achieved by the goal of 'inform flow type details'. The 'flora system provider' depends on 'plantation expert' to achieve goals of 'provider answer' and 'provider flora'. The 'flora system provider' depends on 'news provider' on 'event' and news information. Finally, the 'marketing officer' depends on 'flora system provider' on promotion information.

Figure 6 shows the i* goal model for the goal ‘Ask for Help’, which models the requirements to support real-time enquiry for flora information. Several parties are involved in achieving this goal. They are 'Flora System Provider', ‘Tourism’, 'Network Provider', 'Plantation Expert', and 'Messenger Provider'. In general, ‘Tourism’ depends on 'Flora System Provider' for any flora information. The 'flora system provider' is modelled with the main goal of 'handle help'. In order to achieve this goal, the provider should serve three sub-goals. They are 'send help', 'receive answer' and 'manage message'. The sub-goals and goals are achieved through a plan of 'handle messaging'. During the messaging periods, the 'flora system provider' depends on 'plantation expert' to offer help to ‘Tourism’. In order for this to take place, the 'plantation expert' needs to achieve the main goal of 'handle help'. This

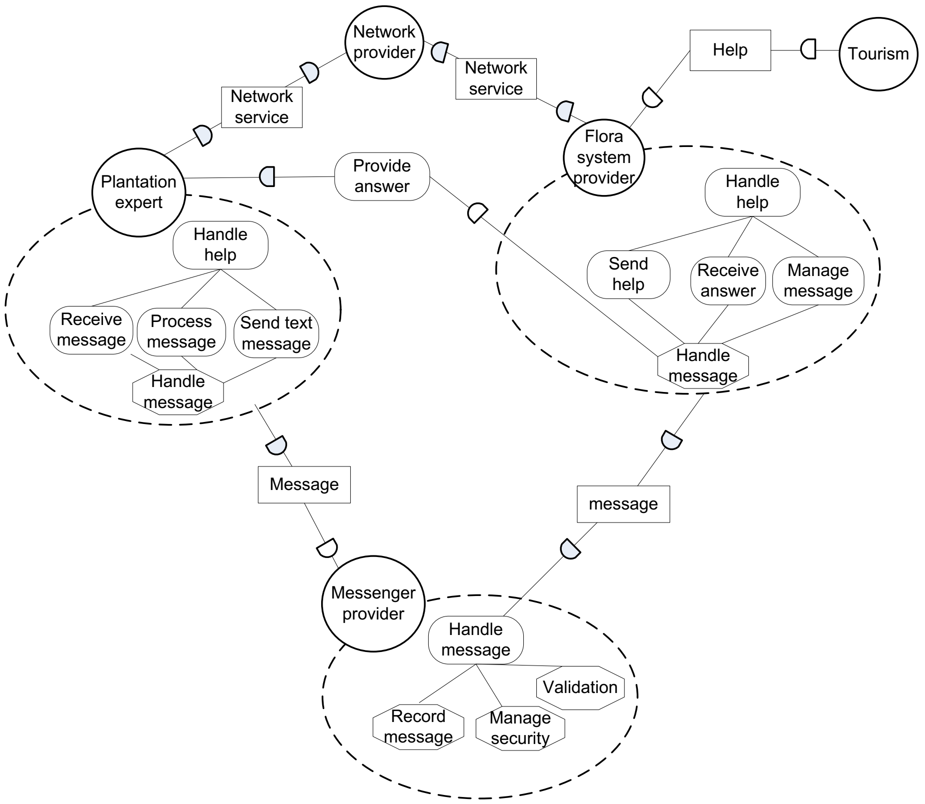


Fig. 6. i* goal model for a goal of 'ask for help'

involves serving the goals of 'receive message', 'process message' and 'send text message'. The messaging requires support from 'messenger provider' and 'network provider' in order to handle the connectivity and messaging service during the enquiry process. The 'messenger provider' will serve the goal of 'handle message', where it is supported by plans like 'record message', 'manage security' and 'validation'. As the actors communicate through messages, a proper user validation is needed to verify the identity of 'Tourism'. This is important to avoid any miscommunication between the tourism and flora provider and ensure better services to customers. Meanwhile, the communication must be done securely in order to ensure trustability and reputability of the system.

As mentioned before, software system requirement is classified into functional and non-functional requirements. Figure 3 presents a higher level abstraction of the system through the ROADMAP goal model. Figure 5 Figure 6 present the functionalities of the system through the i* goal model, showcasing the purpose of the system. Also, Figure 5 and Figure 6 define specific facilities which must be provided by the system.

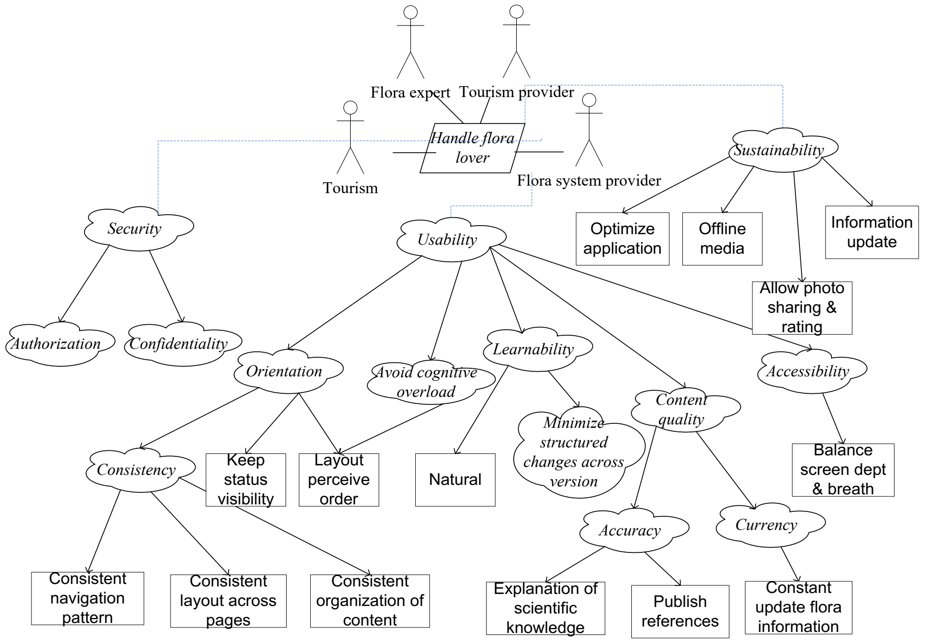


Fig. 7. Non -functional requirements for Kuching FloraLover

Figure 7 presents the non-functional requirement of the Kuching FloraLover application. Figure 7 presents the non-functional requirements for Kuching FloraLover through the i* model extension. The model further details the quality goal from the ROADMAP goal model in Figure 3. As shown in Figure 3, there are three non-functional requirements or quality goals. The aim of the Security quality goal is to avoid disclosure of information to the public. The Usability quality goal is related to user satisfaction. Finally, the Sustainability quality goal ensures the timely usage of the system. In Figure 7, the main quality goals are further analysed through different quality goals. It showcases the decomposition of a quality goal into dedicated quality goals. For example, to achieve the Security quality goal, the other quality goals Authorization and Confidentiality need to be firstly fulfilled. Meanwhile, achieving the Usability goal requires the achievement of other quality goals namely Orientation, Avoid Cognitive Overload, Learnability, Accessibility and Content. Finally, the model is presented with plans that lead to the achievement of the corresponding quality goal. For example, plans like ‘consistent navigation patterns’, ‘consistent layout across pages’ and ‘consistent organised content’ are introduced to ensure the consistency of Kuching FloraLover. Plans such as ‘keep status visibility’ and ‘layout perceived order’ are introduced to achieve the quality goal of ‘orientation’ and ‘avoid cognitive overload’. Meanwhile, plans like ‘natural design’, ‘minimise structured changes across version’ are introduced to improve the learnability of the system. The plan ‘balance screen depth and breath’ is introduced to ensure the accessibility of the system, and ‘explanation of scientific flora change’ and ‘publish references’ are introduced to ensure the accuracy of the system and hence ‘content quality’ of the system. The ‘allow photo sharing and rating’, ensure ‘offline mode’, ‘power management’, and ‘information updating’ are the plans towards the achievement of sustainable Kuching FloraLover. The details of the non-functional requirements for Kuching FloraLover are listed in Table 2.

Table 2. Definition of quality goals for Kuching FloraLover

Authorization: Only a registered user is allowed to send messages, give comments and share their flora lover photo with the system.
Confidentiality: The registration information is kept secure and is not disclosed to the public.
Usability : The quality goal for user satisfaction.
Orientation: The quality goal for the layout of the system.
Consistency: The quality goal for the appearance of the system.
Avoid cognitive overload: The quality goal for the level of user interpretation of the system.
Learnability: The quality goal refers for the ease of learnability of the system, with minimum guideline.
Accessibility: The quality goal for suitability of the system across different levels of users.
Content quality: The quality goal ensuring the good quality of the content.
Accuracy: The quality goal referring to the correctness of content in the system.
Currency: The goal ensuring correct and latest information.
Sustainable: The goal ensuring that the system is always used by ‘Tourism’.

A details non-functional requirement is presented in Figure 8. It presents the dependency analysis among the quality goals through i* model. It further models the actor's involvement in achieving the non-functional requirements. In this case, the 'flora system provider' depends on 'flora expert', 'news provider', 'marketing officer' and 'flora hostage' to provide accurate information. Accurate information is crucial in order to ensure user satisfaction.

‘Tourism’ depends on the 'flora system provider' to ensure secure data transmission. The 'flora system provider' is modelled with various plans to ensure system usability. Finally, the achievement of a sustainability quality goal depends on ‘Tourism’ to share comments regarding the flora itself and rating of flora photos. On the other hand, the system is sustainable if the 'flora system provider' continues to update flora information, event, provide offline management of the system and provide energy optimisation on the functionalities of the system.

Figure 3 to Figure 8 show the modelling of the Kuching FloraLover. In order to validate the models, a paper prototype is developed. Figure 9 shows part of the Kuching FloraLover paper prototype that showcases the preview on the native giant Rafflesia plant. The main screen for the preview consists of three tabs. Tab 1 (left) presents the general information for the plant, whereas Tab 2 presents the Rafflesia’s photo gallery. Tab 3 shows the locality within which the Rafflesia grows. From the paper prototype, we can validate with the end users to ensure the system has fulfilled the user needs. Furthermore, we can transform the paper prototype into implementation for further verification.

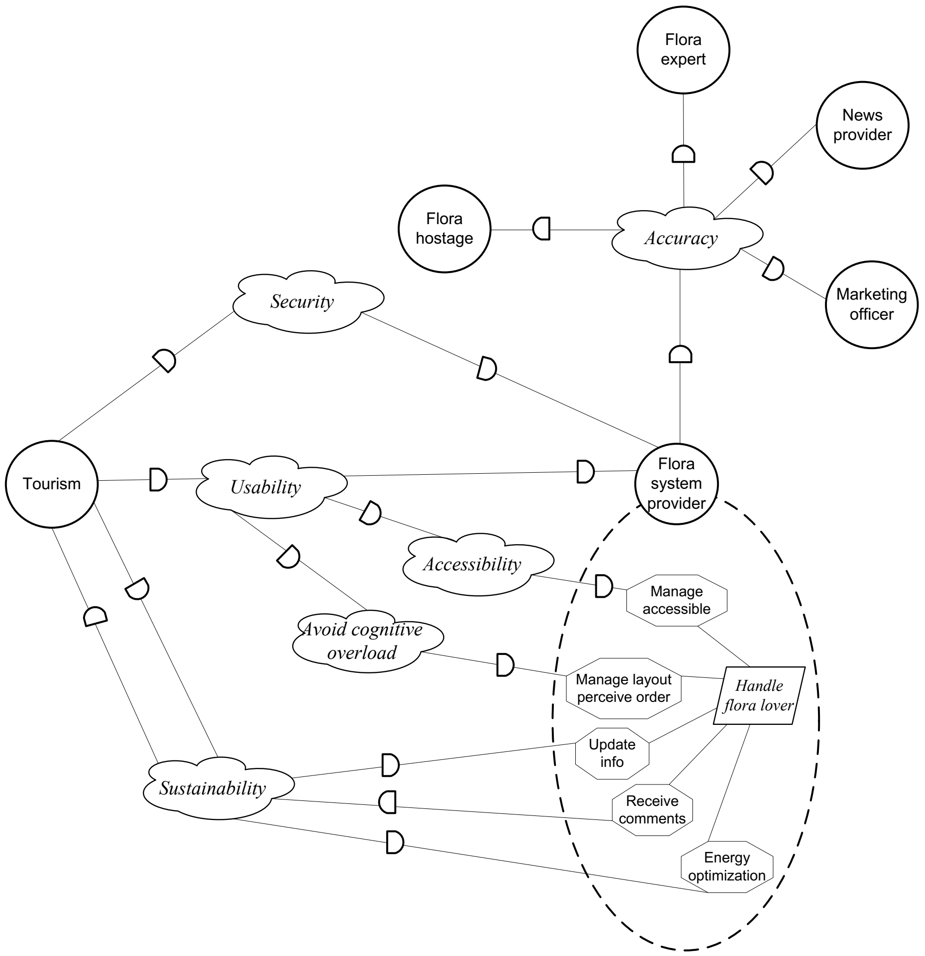


Fig. 8. Quality goal dependency analysis in the i* model

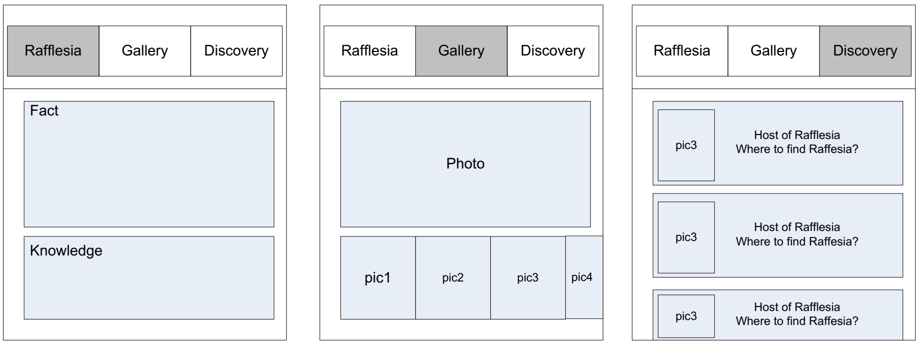


Fig. 9. Paper prototype of Kuching FloraLover

5 Comparing AORE with Current Methods

The details of the proposed AORE for mobile application development are presented in the previous section. Table 3 shows the comparison of the proposed methods with the existing mobile application development method in order to validate our proposed approach qualitatively. The features of the analysis are adopted from [8, 17] in which they represent the characteristics in mobile application development like environment, focus, team size, reliability, application size, time to market, multiple platforms, suitability, architecture, refactoring, users involvement, documentation.

From the Table 3, it is sensible to claim that a hybrid approach is introduced when developing a methodology for mobile application development. For example agile methodology is combined with usability engineering in [9, 4]. Hence, the developer is not provided with a consistent set of models and there is no standard way to build a mobile application. Although various steps have been introduced to hybridise the mobile development methodology, some steps lack detailed descriptions, while some other step lacks a support model. The dependency and the relationship of the steps are also implicit. Interestingly, a more structured method has been introduced through a model-driven approach [5,6,8]. As AORE is based on the AOM methodology, AORE can serve as the initial step to develop a mobile application through Agent Oriented Methodology. In this case, a developer can design a mobile application by transforming the models from AORE as stated in the previous section.

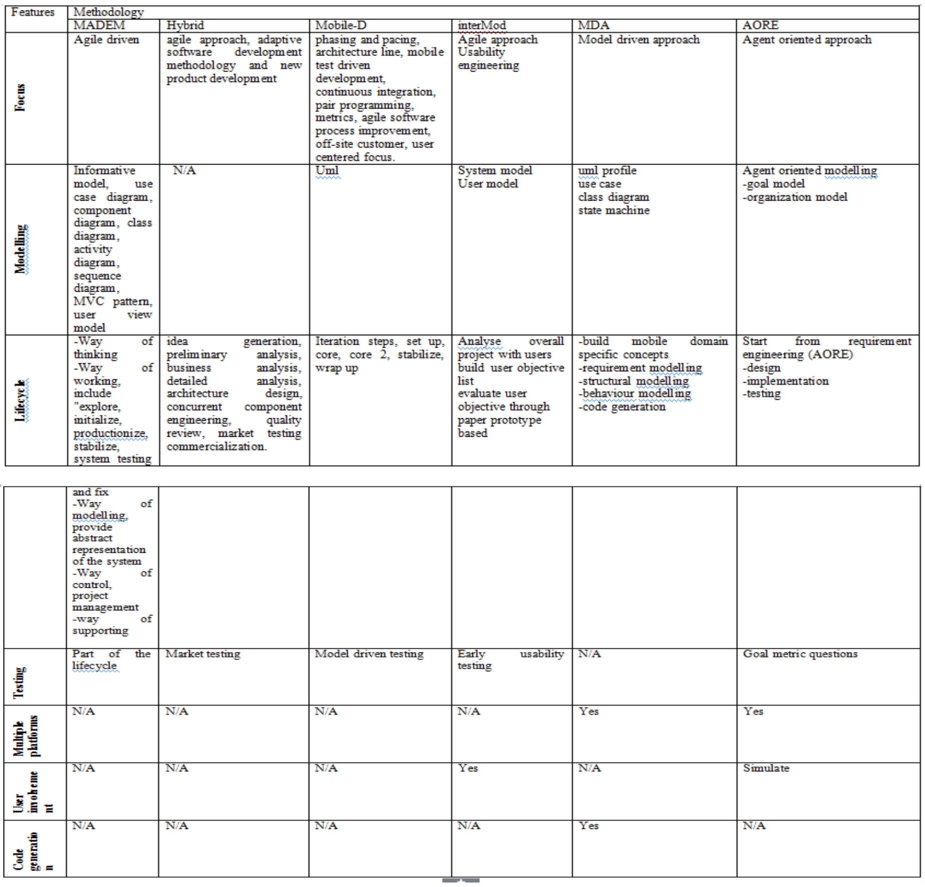
Based on the current practices, the testing step always happens in the last steps of the development. According to [18], user testing is an important step for mobile development due to the changing of user requirements. Hence, the testing should be done at the requirement phase instead of wait until the design and implementation phase. With the AORE, testing can be done through goal modelling and paper prototype. Also, we can transform the goal model into goal metric question [19], a future work to explore.

One important criteria are that software models should able to describe functional and non-functional requirements of the system. As UML is insufficient to model non-functional requirements, there is a rising concern to study non-functional requirements in mobile application development [2]. With AORE, the goal model is highlighted and can serve as a modelling tool to model non-functional requirement.

Another concern for mobile application development is deployed ability on various platforms. This issue, however, it less discussed in most methodologies. As the goal model is conceptual, which represents the mobile application at the abstract level, AORE can be used to design an application for multiple platforms. However, additional work outside the scope of this paper is still needed to validate this claim. User involvement is one important criterion for mobile applications. In reality, though, this is not entirely practical since some users are unsure of what they actually want and also because some might not be interested in participating. In order to close the gap, the goal model presents a network of actors by modelling the dependency between the actors and goals. Indirectly, this closes the gaps allowing better understanding of users.

In all, AORE can serve as an alternative methodology for mobile application development. However, there are some aspects that need to be done in order to validate our claims and we leave it for future study.

Table 3. Comparison of AORE with Current Methods on Mobile Application Development



Features	Methodology					
	MADEM	Hybrid	Mobile-D	interMod	MDA	AORE
Focus	Agile driven	agile approach, adaptive software development methodology and new product development	phasing and pacing, architecture line, mobile test driven development, continuous integration, pair programming, metrics, agile software process improvement, off-site customer, user centered focus.	Agile approach Usability engineering	Model driven approach	Agent oriented approach
Modelling	Informative model, use case diagram, component diagram, class diagram, activity diagram, sequence diagram, MVC pattern, user view model	N/A	Uml	System model User model	uml profile use case class diagram state machine	Agent oriented modelling -goal model -organization model
Lifecycle	-Way of thinking -Way of working, include "explore, initialize, productionize, stabilize, system testing and fix -Way of modelling, provide abstract representation of the system -Way of control, project management -way of supporting	idea generation, preliminary analysis, business analysis, detailed analysis, architecture design, concurrent component engineering, quality review, market testing commercialization.	Iteration steps, set up, core, core 2, stabilize, wrap up	Analyse overall project with users build user objective list evaluate user objective through paper prototype based	-build mobile domain specific concepts -requirement modelling -structural modelling -behaviour modelling -code generation	Start from requirement engineering (AORE) -design -implementation -testing
Testing	Part of the lifecycle	Market testing	Model driven testing	Early usability testing	N/A	Goal metric questions
Multiple platforms	N/A	N/A	N/A	N/A	Yes	Yes
User involvement	N/A	N/A	N/A	Yes	N/A	Simulate
Code generation	N/A	N/A	N/A	N/A	Yes	N/A

6 Related Works

Several methodologies have been introduced for mobile application development. They are mobile-D[1], hybrid agile methodology [4], MADeM [3], intermod [9], MDA [8.9] and Agile [9]. These methodologies were introduced in order to provide structure in supporting rapid changes in stakeholder's requirements and to obtain immediate user feedback [3], as well as to improve effectiveness through reduction of redundant software code. In addition, these methodologies address high-level issues instead of low-level implementation issues [4].

Mobile-D [1] is an agile-based method that relies on UML models. Mobile-D consists of five iterative steps and nine principals, as listed in Table 3. Although UML is adopted as the modelling technique, no clear, detailed description is provided, even for each phase. A lightweight Mobile Application Development Method (MADeM) [3] was proposed based on the principles of methodology engineering framework and provides eight SMART (Simple, meaningful, adequate, realistic and track able) models. MADeM consists of five "ways". They are "way of thinking" that presents the philosophy of the problem and solution; the “way of working” that defines the process of mobile development; “way of modelling” that provides an abstract representation of the system; and “way of control” that focuses on project management. A hybrid methodology was introduced based on the principles of the agile approach, adaptive software development methodology and new product development [4]. However, the details of the methodology are not described in most of the work [4]. Model-driven development was introduced in [5,6,7] to reduce the technical complexity and development costs of mobile application development. In one example namely the GUI model, process models are adopted, but details of the models are not defined. Intermod [9] is based on the agile approach and usability engineering. It focuses on early user testing through questionnaires and paper prototypes. In [6], the MDA approach adopted UML modellings such as class diagram and the state machine diagram for mobile application development. Also, the modeller can generate business logic codes for multiple platforms.

In all, methodologies are needed for mobile application development. Based on the review, current methodologies are not systematic for mobile application development. In line with the MDA approach in [8], we argue that requirements engineering should serve as the first class entity in mobile application development in order for a system to be well-defined. In this paper, the proposed AORE is built on MDA and focuses on agent-oriented modelling. Also, it supports user involvement and early testing through paper prototypes and informal interviews.

7 Conclusion

Several methodologies have been introduced for mobile application development. However, the adoption of agent modelling for mobile application development has no much addressing. This paper explores the potential of agent models like goal model, role model and organisation model to model the requirement of the mobile application. Requirement engineering is an important step in any software application development. With the models, it explicitly describes the functional and non-functional requirements of the mobile application. It showcases a better understanding of a mobile application. A case study and qualitative analysis of the proposed AORE have provided the insight of how agent models can support mobile application development. In future, more works are needed to validate the proposed AORE through case studies and empirical studies.

8 Acknowledgement

The funding for this project is made possible through the research grant obtained from UNIMAS under the Yayasan Dayak Chair [Grant No: F08/DRC/1458/2016] entitled, Mobile Flora information system for Visitor in Kuching division.

9 References

[1] Abrahamsson, P., Hanhineva, A., Hulkko, H., Ihme, T., Jalinoja, J., Korkala, M., ... & Salo, O. (2004). Mobile-D: an agile approach for mobile application development. 19th Annual ACM SIGPLAN Conference on Object-Oriented Programming Systems, Languages, and Applications , pp 174-175 https://doi.org/10.1145/1028664.1028736

[2] Stapić, Z., Mijač, M., & Strahonja, V. (2016). Methodologies for development of mobile applications. 39th International Convention on Information and Communication Technology, Electronics and Microelectronics (MIPRO). pp. 688-692 https://doi.org/10.1109/MIPRO.2016.7522228

[3] Alsabi, E., & Dahanayake, A. (2016). Smart Modeling for Lightweight Mobile Application Development Methods. East European Conference on Advances in Databases and Information Systems. pp. 168-179 https://doi.org/10.1007/978-3-319-44066-8_18

[4] Rahimian, V., & Ramsin, R. (2008). Designing an Agile Methodology For Mobile Software Development: A Hybrid Method Engineering Approach. International Conference on Research Challenges in Information Science. pp 337-342IEEE. https://doi.org/10.1109/RCIS.2008.4632123

[5] Vaupel, S., Taentzer, G., Gerlach, R., & Guckert, M. (2016). Model-driven development of mobile applications for Android and iOS supporting role-based app variability. Software & Systems Modeling. pp 1-29 https://doi.org/10.1007/s10270-016-0559-4

[6] Usman, M., Iqbal, M. Z., & Khan, M. U. (2014). A model-driven approach to generate mobile applications for multiple platforms. 21st Asia-Pacific Software Engineering Conference,1: 111-118 https://doi.org/10.1109/APSEC.2014.26

[7] Nagappan, M., & Shihab, E. (2016, March). Future trends in software engineering research for mobile apps. In 2016 IEEE 23rd International Conference on Software Analysis, Evolution, and Reengineering (SANER),5: 21-32 https://doi.org/10.1109/SANER.2016.88

[8] Umuhoza, E., & Brambilla, M. (2016). Model driven development approaches for mobile applications: A survey. International Conference on Mobile Web and Information Systems. pp 93-107 https://doi.org/10.1007/978-3-319-44215-0_8

[9] Losada, B., Urretavizcaya, M., López-Gil, J. M., & Fernández-Castro, I. (2012, October). Combining InterMod agile methodology with usability engineering in a mobile application development. 13th International Conference on Interacción Persona-Ordenador .pp 39 https://doi.org/10.1145/2379636.2379674

[10] Pressman, R. S. (2005). Software engineering: a practitioner's approach. Palgrave Macmillan.

[11] Juan, T., Pearce, A., & Sterling, L. (2002). ROADMAP: extending the gaia methodology for complex open systems. International Joint Conference on Autonomous Agents and Multiagent Systems: Part 1. pp. 3-10 https://doi.org/10.1145/544741.544744

[12] Shiang, C. W., Halin, A. A., Lu, M., & CheeWhye, G. (2016). Long Lamai Community ICT4D E-Commerce System Modelling: An Agent Oriented Role-Based Approach. The Electronic Journal of Information Systems in Developing Countries, pp 75

[13] WaiShiang, C., YeeWai, S., Nizam, S., & CheeWyai, L. (2016). Agent oriented requirement engineering for lake mathematical modelling: Preliminary study. Journal of Telecommunication, Electronic and Computer Engineering (JTEC), 8(2), 5-10.

[14] Shiang, C. W., Onn, B. T., Tee, F. S., bin Khairuddin, M. A., & Mahunnah, M. (2016). Developing Agent-Oriented Video Surveillance System through Agent-Oriented Methodology (AOM). CIT. Journal of Computing and Information Technology, 24(4), 349-368. https://doi.org/10.20532/cit.2016.1002869

[15] Cheah, W. S., Mit, E., & Abdul Halin, A. (2015). Shared single display application: an interactive patterns approach. International Journal of Software Engineering and Its Applications, 9(2), 233-250.

[16] CheeWyai, L., Cheah, W., Chowdhury, A. K., & Gulden, C. (2015). Engineering sustainable software: A case study from offline computer support collaborative annotation system. 9th Malaysian Software Engineering Conference (MySEC),. pp 272-277 https://doi.org/10.1109/MySEC.2015.7475232

[17] Kaur, A., & Kaur, K. (2015). Suitability of Existing Software Development Life Cycle (SDLC) in Context of Mobile Application Development Life Cycle (MADLC). International Journal of Computer Applications, 116(19). https://doi.org/10.5120/20441-2785

[18] Reis, R. A. C., Fontão, A. D. L., Gomes, L. L., & Dias-Neto, A. C. (2015). Usability Evaluation Approaches for (Ubiquitous) Mobile Applications: A Systematic Mapping Study. International Conference on Mobile Ubiquitous Computing, Systems, Services and Technologies. UBICOMM.

[19] Van Solingen, R., Basili, V., Caldiera, G., & Rombach, H. D. (2002). Goal Question Metric (GQM) approach. Encyclopedia of Software Engineering. Chicago. https://doi.org/10.1002/0471028959.sof142

10 Authors

WaiShiang Cheah is a senior lecturer at the Faculty of Computer Science and Information Technology, Institute of Borneo Studies, Universiti Malaysia Sarawak, Malaysia.

Aida. B. is a senior lecturer at the Faculty of Resource Science & Technology, Department of plantation, Universiti Malaysia Sarawak, (apashafreena@unimas.my).

Puah T. C. is an associate professor at the Faculty of Economic, Universiti Malaysia Sarawak, Malaysia (chpuah@unimas.my).

Alfian Abdul Halim is a senior lecturer at the Faculty of Computer Science, Universiti Putra Malaysia, Selangor, Malaysia (e-mail: alfian@ieee.org).

Article submitted 10 February 2017. Published as resubmitted by the authors 12 April 2017.