An Enhancement of Software Requirements Negotiation with Rule-based Reasoning: A Conceptual Model

Sabrina Ahmad, Intan Ermahani A. Jalil, Sharifah Sakinah Syed Ahmad
Faculty of Information and Communication Technology,
Universiti Teknikal Malaysia Melaka, Hang Tuah Jaya, 76100 Durian Tunggal, Melaka, Malaysia.
sabrinaahmad @utem.edu.my

***Abstract*—Problems to be solved by software systems are becoming complex and the requirements of these systems are based on increasingly detailed knowledge of the users' domain. Stakeholders are people related to the system and usually come to the elicitation process with different perception and perspective. Dealing with multiple stakeholders, conflicts are inevitable and therefore the need of negotiation mechanism to resolve conflict is crucial. This paper forwards an enhancement of software requirements negotiation conceptual model to assist the conflict detection and resolution effort. The significant of the enhanced model is to empower the automation of conflicts detection and its severity level with rule-based reasoning.**

***Index Terms*—Software Requirements; Negotiation; Human Factors; Conflicts Detection and Resolution; Requirements Engineering; Rule-Based Reasoning.**

I. INTRODUCTION

Software requirements are a foundation to the software development. It is important as it mould the shape of the software, acknowledge the cost and the duration of the development, describe the functionalities of the system, clarify the system constraints and discover the quality attributes it must possess. Hence, the effect of poor software requirements is severe. The effects include cost rework, budget overruns, poor quality system, stakeholders' dissatisfaction and project failures [4]. However, the effort to produce quality requirements is scarce.

Nowadays, problems to be solved by software systems are becoming complex and the requirements of these systems are based on increasingly detailed knowledge of the stakeholders' domain. Stakeholders are people related to the system and affected by the system directly or indirectly. Also, stakeholders usually have valid interest in the system that come to the elicitation process with different perception and perspective [16]. Dealing with multiple stakeholders, conflicts are inevitable [14]. It is seldom technical difficulties which inhibit the process of requirements elicitation but mostly are human factors and subjectivity [12]. Based on a survey in China [11], it is reported that the major failure of requirements engineering practices are traced back to the stakeholders such as customers' lack of understanding on the system requirements themselves and the users' needs and understanding constantly change throughout the process. These understanding problems among the multiple stakeholders usually lead to conflicts. It is reported that conflicts among the stakeholders highly influence the project success factor in public sectors in Malaysia [16].

Hence, this paper presents an enhanced software requirements negotiation conceptual model. The enhanced model introduced several features to promote simplification and automation. Following Introduction, Section 2 presents the background. This is followed by Section 3 that presents the improved model. Next, Section 4 presents the role of rule-based reasoning. Then, Section 5 describes the enhancement features of the negotiation model and Section 6 concludes the paper.

II. BACKGROUND

In a process of identifying the right requirements to develop, conflicts are common since stakeholders frequently pursue mismatching goals. Reaching agreements among stakeholders who have different concerns, responsibilities, and priorities is quite challenging. Inspired by Theory W, Barry Boehm introduced WinWin Model to handle stakeholders' dispute [2]. This model realizes a negotiation effort to handle conflicts and to resolve disagreement.

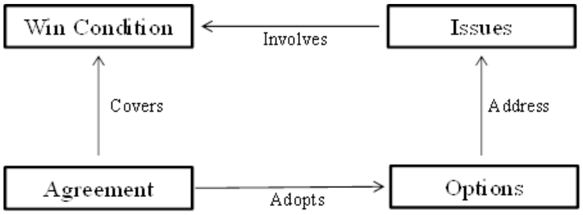

Figure 1: WinWin Model

Referring Figure 1, the negotiation model guides success critical stakeholders in elaborating mutually satisfactory agreements. Stakeholders express their goals as win conditions. If everyone agrees, the win conditions become agreements. When stakeholders do not agree, they identify their conflicted win conditions and register their conflicts as issues. In this case, stakeholders invent options for mutual gain and explore the option trade-offs. Options are iterated and turned into agreements when all stakeholders agree [2]. Several generations of WinWin groupware has been developed to assist the implementation of the WinWin Model. Table 1 summarized the evolution of negotiation efforts.

Table 1
Software Requirements Negotiation Efforts

Year	Author/s	Article Title	Contributions
2001	P. Grünbacher and B. Boehm	EasyWinWin: A groupware-supported methodology for requirements negotiation	EasyWinWin -A groupware to allow group interaction to perform software requirements negotiation in order to resolve conflicts.
2003	P. Grünbacher and P. Braunsberger	Tool Support for Distributed Requirements Negotiation	ARENA – A web-based tool to allow distributed requirements negotiation.
2008	D. Yang	Wikiwinwin: A wiki based system for collaborative requirements negotiation	WikiWinWin – Improved EasyWinWin by embedding Wiki Technology
2012	N. Kukreja	WinBook: A social networking based Framework for Collaborative Requirements Elicitation and WinWin Negotiations	WinBook – Improved WikiWinWin by improving the usability of non-technical users.
2012	A. Felfernig, et al.	Group Decision Support for Requirements Negotiation	IntelliReq – Embed recommendation technologies to assist in group decision.

In evolution to the WinWin Model, a groupware called EasyWinWin is developed [7]. It is based on a Group Support System (GSS) which is a suite of software tools that can create, sustain, and change patterns of group interaction in repeatable, predictable ways. The GSS is meant to provide platform for group discussion as any user can make a contribution to a shared list and any contribution a person makes appears instantly on all the other users' screens. The EasyWinWin is seen as a successful negotiation tool to reveal and to resolve conflicts. In 2003, ARENA (Anytime, Anyplace REquirements Negotiation Aids) is developed. It is a web based tool which is based on EasyWinWin that allowed distributed and asynchronous requirements negotiation [7]. Later, ARENA-M was developed to allow mobile stakeholders to participate in requirements negotiation. Embedding Wiki Technology, WikiWinWin was developed in 2008 to empower EasyWinWin [15]. The Wiki approach was seen easier to learn and use, more flexible and easy to update requirements in order to organize information. Later in 2012, WinBook was introduced [10]. The WinBook combined Gmail and Facebook technologies to bring forward a more user-friendly tool for non-technical users for simplification. Besides, a research [5] embed recommendation technologies to assist in group decision and developed a tool called IntelliReq. It was designed to support group decision process in small sized software project (6-8 team members). Empirical investigation was conducted and the result shown that it improved the perceived usability and quality of decision support. The improved new model which will be presented in Section 3 is based on the WinWin Model. The capabilities of the enhanced model are proposed after considering previous efforts.

III. The Improved Model

In a process of identifying the right requirements to develop, conflicts are common since stakeholders frequently pursue mismatching goals. Reaching agreements among stakeholders who have different concerns, responsibilities, and priorities is quite challenging. Inspired by Theory W, Barry Boehm introduced WinWin Model to handle stakeholders' dispute [2]. This model realizes a negotiation effort to handle conflicts and to resolve disagreement. Motivated by WinWin Model, an improved model is proposed. Comparable to the WinWin Model groupware and evolutions, the new model allows participated stakeholders to share the glossary used throughout the process in order to promote mutual understanding. The stakeholders are also responsible to register the candidate requirements which will be viewed by all the participating stakeholders. In order to understand the stakeholders' needs on the system to be developed, the stakeholders have to register individual preferences for every candidate requirements. Then only the system will automatically detect the conflicts and at the same time prioritize the conflicts severity level. The conflicted candidate requirements will be translated into issues which need stakeholders' attention to resolve. In order to facilitate the resolution process, the stakeholders are given opportunity to add rational of the requirements and the importance of having the specific requirements. If the provided knowledge is still not sufficient to converge, stakeholders can register options as new candidate requirements. The process will iterate until total consensus is achieved.

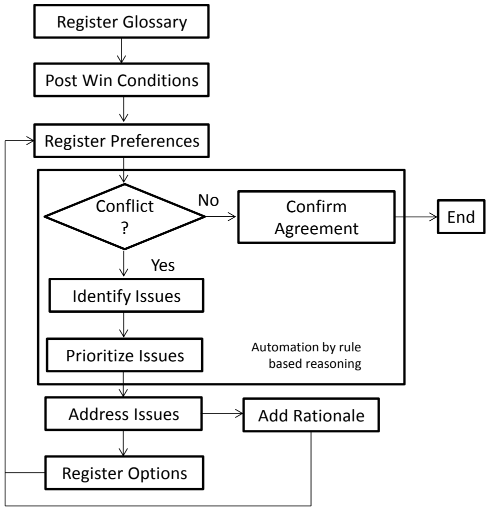

Figure 2: The Improved New Model to Detect and to Resolve Conflicts with Rule-based Reasoning

Referring to Figure 2, the more detail process flow of the model is explained as below:

Register Glossary: The model starts with all the stakeholders register their glossary. Glossary is to define the meaning of specific terms which may not often used by all the participating stakeholders. This is to allow mutual and common understanding as the glossary is used as a reference for the stakeholders to understand the knowledge throughout the process.

Post Win Conditions: Next, all the stakeholders post their candidate requirements (known as win conditions) within the time permitted. The time boundary allowed can be adjusted and must be agreed among all the participated stakeholders. Each time the win condition is posted, it will appear in everyone's screens.

Register Preferences: When all the candidate requirements are listed, the stakeholders have to register their preferences. The preferences are based on scale 0-4 which is adapted from prioritization method [1]. MoSCoW is a prioritization technique used in business analysis and software development to reach a common understanding with stakeholders on the importance they place on the delivery of each requirement. The capital letters in MoSCoW stand for M - MUST have this, S - SHOULD have this if at all possible, C - COULD have this if it does not affect anything else and W - WON'T have at this time but WOULD like in the future.

This method was converted into a numbered scale from 0 to 4 in which an item was added to scale 0 meaning 'Must never have this.' This item was introduced to provide an option if the stakeholders do not want the particular requirement to be included. This is possible in a circumstance of requirements which are requested by a stakeholder but is not wanted by the other. Table 2 below state the scale used to register preferences in the model.

Table 2
The Scale for Preferences Prioritization

Scale	Meaning
4	Must have this
3	Should have this if at all possible
2	Could have this if it does not affect anything else
1	Will not have this time but would like in the future
0	Must never have this

Conflict: Then the model will automatically detect the existence of conflicts if any. Refer Section 4 for more details.

Identify Issues: If any conflict is detected, then the conflict will be classified as issues which will be highlighted for the stakeholders to resolve.

Prioritize Issues: Whenever the issues are identified, the prioritization of the issues severity level will be automatically recognized as well. The prioritization will be divided into high, moderate and low severity level which gives the stakeholders indication of importance (Refer Section 4).

Address Issues: In order to address issues, the stakeholders can either add rationale to persuade others to achieve agreement or register options for further consideration.

Adds Rationale: For each issue, the stakeholders can add in rationale to provide information or to explain the importance of the candidate requirements. At the same time, the stakeholders are also allowed to justify the voting value they registered.

Register Options: If explanation and persuasion failed, other options will be registered. The exploration of options will go through the same iteration as new candidate requirements and the stakeholders can register their preferences.

Confirms Agreement: If there is no conflict, then agreement is achieved. The model will let much iteration to register preferences until total agreement is succeeded.

IV. THE ROLE OF RULE-BASED REASONING

The rule-based reasoning method is a famous technique used in an expert system. There are several other domains that already gain benefit from this method which includes the research area of medical, pattern recognition, transportation and also marketing industry [3,15,9,6]. In the identification of each problems in any domain, the experts' view are the most important to assemble the set of rules for the inference engine. This method is suitable for any problem that accepts several inputs and having to produce several outputs or solutions.

The basic rule presentation is as below:

```
If condition
        Then action
```

One condition or a series of conditions that explains certain amount of data or situations that need to be fulfilled belong to 'if' part while the 'then' part is responsible to set an action that need to be done or to propose any solutions once the conditions are accepted. In order to automate the conflict detection, this process depends on rule-based reasoning module to determine whether there are any conflicts among the stakeholders for any candidate requirements. Figure 3 shows the rule-based reasoning model with inference engine for conflict detection.

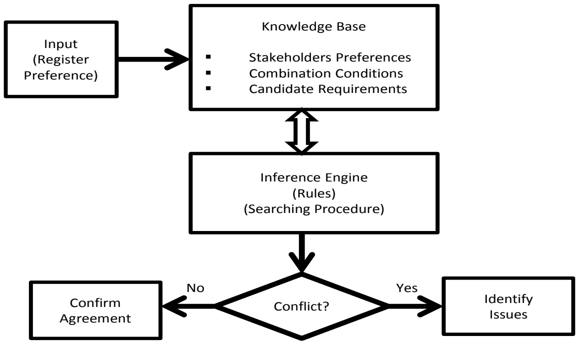

Figure 3: Rule-based reasoning model with Inference Engine for Conflict Detection

Here, the experts' view is referred to the stakeholders' input based on their registered preferences. The inference engine will store the rules for every condition exists to determine either conflicts or none conflicts state. The situation can get more complicated if there are any situations with no matching conditions or solutions. In this case, the inference engine need to be updated with all possibilities of stakeholders different choices of preferences that can led towards two state either it will set out to conflict or none conflict situation. The conflict situation is then leads to determination of each conflicts severity level.

Severity level for any conflict situations are based on how rigorous each stakeholder's preferences from the others and resulted from the automation process of negotiation for each candidate requirements. Thus, conflicted situation is diverged into three severity levels consist of HIGH, MODERATE and LOW level of conflict situations. Table 3 presents an example of solution summary for some conflicts and its severity level based on five candidate requirements (win condition) elicited from a case study named Academic Unit Registration System. The severity levels were identified based on five stakeholders (S1 until S5) who registered their preferences based on scale 0-4.

Table 3
An example of solution summary for some conflicts and its severity level based on five candidate requirements

No	Requirement	S1	S2	S3	S4	S5	Severity Level
1	Retrieve information about the unit offer for the current semester.	0	1	2	3	4	High
2	Create schedule by selecting and registering four units with two extra choices from the catalogue.	4	4	3	0	0	High
3	Pay tuition fee online.	1	1	4	4	4	Moderate
4	Create student's schedule for the semester and allow modification throughout the semester.	2	2	4	4	4	Moderate
5	Notify students by electronic mail once the schedules have been processed.	2	2	3	3	3	Low

The automation process via rule based reasoning model requires complete development of rules for its inference engine. Two techniques are used to generate the rules; by using Mockler Chart and Decision Table. The following explanation is based on some example of solution summary listed in Table 3.

A. Mockler Chart

Figure 4 and 5 shows the Mockler Chart that presents an overview to the input from each stakeholders' preferences, some rules generated based on input combinations and the suggestion made by the chart whether all stakeholders' preferences will contribute towards any conflict and the severity level that regards to it or no conflict at all. The chart also presents the relationship among stakeholders' preferences. The chart models that when all stakeholders choose difference choices based on their preferences towards a certain requirement, the severity level for their preferences is set to HIGH in conflict matters as shown in Figure 4. Besides, as shown in Figure 5 when all stakeholders choose the same choices will result to NO conflict at all.

B. Decision table

Decision table which correspond towards the action or solution that are needed to detect whether there exist any possible conflicts is shown in Table 4. This marks the combination of preferences for all stakeholders that resulted towards the solutions for the severity level. As shown in the condition part, the Y shows the selected preferences for each stakeholder based on the scale 0-4 for preferences. Each column represents an example of the combination of each stakeholder preferences' selection for a certain requirement. The X marks the solution result (an action of either NO conflicts or Conflict with HIGH, MODERATE or LOW level) based on the combinations of each stakeholders' preferences.

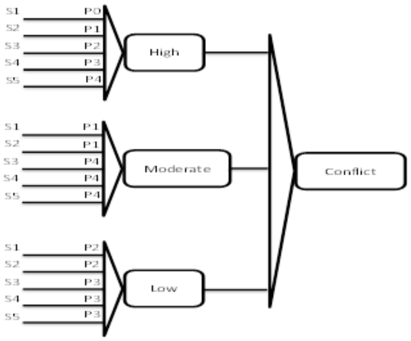

Figure 4: Mockler Chart with decision leads to Conflict; S1, S2, S3, S4 and S5 are referred to Stakeholders; P0, P1, P2, P3 and P4 is referred to preferences

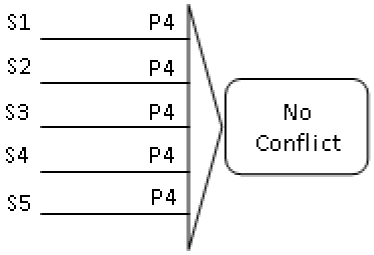

Figure 5: Mockler Chart with decision leads to No Conflict; S1, S2, S3, S4 and S5 are referred to Stakeholders; P4 is referred to preferences

Table 4
Decision Table for Conflicts Detection; S-Stakeholders, P-Preferences

S	P	AND							
Condition									
1	0	Y		Y					
1	1				Y	Y			
1	2						Y	Y	
1	3								Y
1	4		Y						
2	0	Y							
2	1			Y		Y			
2	2				Y		Y	Y	
2	3								Y
2	4		Y						
3	0	Y							
3	1								
3	2			Y					
3	3				Y	Y		Y	
3	4		Y				Y		Y
4	0	Y							
4	1								
4	2								
4	3			Y		Y		Y	
4	4		Y		Y		Y		Y
5	0	Y			Y				
5	1								
5	2								
5	3					Y		Y	
5	4		Y	Y			Y		Y
Action									
NONE		X	X						
HIGH				X	X				
MODERATE						X	X		
LOW								X	X

C. Rules for Inference Engine

The rules generated for inference engine are shown in Table 5. The inference engine is responsible to store all possible rules that match the state of either any conflict is detected or there is no conflict to a certain stakeholders' preferences.

If all stakeholders have a mutual understanding towards any candidate requirements either they mutually agreed to accept or not accepting the requirements, then there will be no conflict detected. This can only happen when all stakeholders have the same preference prioritization towards certain candidate requirements. This is shown in the first left column of Table 5.

If all stakeholders have different choices, then there will be a conflict detected. This can happen when all stakeholders have different preference prioritization towards certain candidate requirements. Also when some stakeholders agree on some requirements but some stakeholders don't. This is shown in the first row right column and both column in second row of Table 5.

These rules will be generated and kept in the inference engine. All rules are based on the stakeholders' preferences towards all candidate requirements. The automation of conflict detection will search through all rules in the inference engine to match with the stakeholders' combination of choices.

Table 5
Rules for Inference Engine

IF S1 choose 4 AND S2 choose 4 AND
S3 choose 4 AND S4 choose 4 AND
S5 choose 4 AND
THEN No Conflict
	IF S1 choose 0 AND S2 choose 1 AND
	S3 choose 2 AND S4 choose 3 AND
	S5 choose 4 AND
	THEN Conflict AND
	Severity Level is High

IF S1 choose 2 AND S2 choose 2 AND S3 choose 4 AND S4 choose 4 AND S5 choose 4 AND THEN Conflict AND Severity Level is Moderate	IF S1 choose 2 AND S2 choose 2 AND S3 choose 3 AND S4 choose 3 AND S5 choose 3 AND THEN Conflict AND Severity Level is Low

V. IMPROVEMENT AND ENHANCEMENT FEATURES

The new model has the essential ability of the WinWin Model. It provides a mechanism to detect and to resolve conflicts in a web based platform. Other than allowing the registration of glossary, candidate requirements, preferences and explanation to achieve total agreement in several iterations, explained below are new features incorporates in the new model to boost the performance 5.1

A. Support Global Requirements Engineering.

The new model will come with time boundary mechanism for registering the candidate requirements and the preferences. The time boundary is important to guard the process flow as one process in the model is a pre-requisite to the other. This mechanism will allow distributed stakeholders in different time zone from all over the world to provide sufficient input within agreed time constraint. Embedding time boundary, the participating stakeholders do not have to be in front of their screens at the same time. Still, the negotiation process happens.

B. Provide Guided Walkthrough

The guided walkthrough of the system is to facilitate the usage of the model throughout the process. The stakeholders will be acknowledged on what need to be done, what will happen and how to move forward. In addition, the notification will be sent to the stakeholders as an alert mechanism to take care of the time boundary and to ensure smooth sailing process.

C. Automate Conflicts Detection

The automation of conflicts detection is to handle the process of reviewing each stakeholder's preferences and determine whether their choices will lead towards resulting to conflicts or none conflicts scenarios based on the rule-based reasoning module. The stakeholders will be able to view which candidate's requirements that have conflicts and either revising their choices based on the following process of identify and prioritize issues with the next step requires some stakeholders to make a changes towards their preferences based on any rational added. The agreement is achieved if there is no conflict at all.

D. Automate Severity Level

The Severity Level of any conflict is automatically determined by the rules that are stored in the inference engine. The levels included High level that represents a situation where none of the stakeholders chose the same preference, Moderate level where some group of stakeholders might chose the same but there is a certain gap in scale of prioritization in their choices and lastly Low level where the stakeholders chose almost in near scale of prioritization for their preferences towards the candidate requirements. These severity levels provide the result for conflicted states of candidate requirements.

VI. CONCLUSION

As a conclusion, the enhanced conceptual model is introduced to improve the performance of the current initiatives by embedding rule-based reasoning module. Besides, the overall process flow is improved to simplify the conflicts detection and resolution effort to be applicable for both technical and non-technical users. This is aimed to benefit the requirement elicitation process in reviewing the software requirements.

ACKNOWLEDGMENT

The background of this research work is funded by Fundamental Research Grant Scheme numbered FRGS/2/2013/ICT01/UTEM/02/1.

REFERENCES

[1] MoSCoW prioritisation. Reducing Your Acceptance Testing Risk. Coley Consulting. 2007.

[2] B. Boehm., P. Bose., E. Horowitz. & M. J. Lee. "Software requirements as negotiated win conditions." *First International Conference on*

Requirements Engineering. Colorado Springs, IEEE Computer Society. 1994.

[3] R. Cucchiara., M. Piccardi. & P. Mello. "Image analysis and rule-based reasoning for a traffic monitoring system." IEEE Transactions on Intelligent Transportation Systems, 1, 119-130. 2000.

[4] A. Davis. Just enough requirements management: where software development meets marketing, Colorado, Addison-Wesley. 2013.

[5] A. Felfernig., C. Zehentner., G. Ninaus., H. Grabner., W. Maalej., D. Pagano., L. Weninger. & F. Reinfrank. "Group Decision Support for Requirements Negotiation." IN ARDISSONO, L. & KUFLIK, T. (Eds.) Advances in User Modeling. Springer Berlin Heidelberg. 2012.

[6] D. Gorgevik., D. Cakmakov. & V. Radevski. "Handwritten digit recognition by combining support vector machines using rule-based reasoning." *Proceedings of the 23rd International Conference on Information Technology Interfaces*. Pula, Croatia. 2001.

[7] P. Grünbacher. & B. Boehm. "EasyWinWin: A groupware-supported methodology for requirements negotiation." *8th European Software Engineering Conference held jointly with 9th ACM SIGSOFT International Symposium on Foundations of Software Engineering.* Toronto, Canada. 2001.

[8] P. Grünbacher. & P. Braunsberger. "Tool Support for Distributed Requirements Negotiation." IN CIMITITLE, A., DE LUCIA, A. & GALL, H. (Eds.) Cooperative Methods and Tools for Distributed Software Processes. Milano, Italy, FrancoAngeli. 2003.

[9] A. Hanemann. "A hybrid rule-based/case-based reasoning approach for service fault diagnosis." *20th International Conference on Advanced Information Networking and Applications.* Vienna, Austria. 2006.

[10] N. Kukreja. "WinBook: A social networking based Framework for Collaborative Requirements Elicitation and WinWin Negotiations." *34th International Conference on Software Engineering.* Zurich, Switzerland, IEEE Press. 2012.

[11] L. Lin., L. Tong. & P. Fei. "Why Requirements Engineering Fails: A Survey Report from China". *18th IEEE International Requirements Engineering Conference (RE)*. 2010.

[12] M. E. R. F. Lopes. & C. H Q. Forster. "Application of human error theories for the process improvement of Requirements Engineering." Information Sciences, 250, 142-161. 2013.

[13] D. Rossille., J. F. O. Laurent. & A. Burgun. "Modelling a decision-support system for oncology using rule-based and case-based reasoning methodologies." *International Journal of Medical Informatics*, 74, 299-306. 2005.

[14] N. Rozanski. & E. Woods. Software systems architecture: working with stakeholders using viewpoints and perspectives, Addison-Wesley. 2011.

[15] D. Yang., D. Wu., S. Koolmanojwong., A. Winsor Brown. & B. W. Boehm. "Wikiwinwin: A wiki based system for collaborative requirements negotiation." *41st Annual Hawaii International Conference on System Sciences,* IEEE. 2008.

[16] N. H. Zakaria., A. Haron., S. Sahibuddin. & M. Harun. "Requirement Engineering Critical Issues in Public Sector Software Project Success Factor." *International Journal of Information and Electronics Engineering (fJ1EE)*, 201, 200 - 209. 2011.