A Pair-Oriented Requirements Engineering Approach for Analysing Multi-lingual Requirements

Massila Kamalrudin[1,*], Safiah Sidek[1], Norsaremah Salleh[2], John Hosking[3], and John Grundy[4]

[1]Innovative Software System & Services Group, Universiti Teknikal Malaysia Melaka, Melaka, Malaysia
(massila, safiahsidek)@utem.edu.my
[2]Department of Computer Science, International Islamic University Malaysia, 50728 Kuala Lumpur, Malaysia
norsaremah@iium.edu.my
[3]College of Engineering & Computer Science, Australian National University, Canberra, ACT 0200, Australia
john.hosking@anu.edu.au
[4]Centre for Computing and Engineering Software Systems, Swinburne, University of Technology, PO Box 218, Hawthorn, Victoria 3122, Australia
jgrundy@swin.edu.au

Abstract. Requirements written in multiple languages can lead to error-proneness, inconsistency and incorrectness. In a Malaysian setting, software engineers are exposed to both Malay and English requirements. This can be a challenging task for them especially when capturing and analyzing requirements. Further, they face difficulties to model requirements using semi-formal or formal models. This paper introduces a new approach, Pair-Oriented Requirements Engineering (PORE) that uses an Essential Use Case (EUC) model to capture and analyze multi-lingual requirements. This approach is intended to assist practitioners in developing correct and consistent requirements as well as developing teamwork skills. Two quasi-experiment studies involving 80 participants in the first study and 38 participants in a subsequent study were conducted to evaluate the effectiveness of this approach with respect to correctness and time spent in capturing multi-lingual requirements. It was found that PORE improves accuracy and hence helps users perform better in developing high quality requirements models.

Keywords: Pair-Oriented, Analysing Requirements, Multi-lingual requirements

1 Introduction

Multi-lingual communication is common in countries that have a mother-tongue other than the pervasive English language. In Malaysia, whose primary language is the Malay language, "code-switching" between English and Malay has become a common practice of communication [1]. Code-switching has also become a common prac-

adfa, p. 1, 2011.

tice in the Malaysian IT industry. Considering that the Malay language is the official language of Malaysia, this language is commonly used in the government IT sector, especially in writing the requirements of a system, and when eliciting or capturing requirements from clients or stakeholders who are not fluent in English. Meanwhile, the English language is commonly used in the business IT sector. This situation leads to a plethora of multi-lingual requirements – those expressed in both Malay and English languages, or a mixture of both [1]. Working with multi-lingual requirements, software engineers need to be proficient in both languages to be able to capture quality requirements that meet the needs of the stakeholders.

There are a wide variety of methods for modelling and analysing software requirements. These include goal-oriented, viewpoint-oriented, agent-oriented and object-oriented approaches [2-4]. Although the benefits of these methods are widely recognised, software engineers need a high level of understanding and skill to be able to capture and analyse requirements. As such, besides dealing with multi-lingual issues, software engineers also face difficulties to handling various tools for analysing requirements used in the IT industry.

Considering the use of both Malay and English language in the Malaysian IT industry, students of software engineering in the Malaysian institutions of higher education are exposed to multi-lingual requirements. These students need to be familiar with IT terms and scenarios in both languages so that when they enter the IT industry, they will be able to function effectively in this environment. Further, it has been reported that many students have trouble in capturing requirements using semi-formal or formal models such as UML, tabular models, algebra or mathematics [5].

Motivated by these issues, we propose a new approach for requirements capture and analysis, Pair-oriented Requirements Engineering (PORE). We expect this approach to be suitable for novice users who in this case face two main issues in software development. The first issue relates to knowledge and skills of the various methods to analyse requirements, and the second issue relates to the analysis of multi-language requirements. The PORE approach builds on our earlier work to support the development and analysis of multi-lingual requirements using the EUC modelling approach [1]. In our earlier work, we have developed a new toolset for developing and evaluating EUCs in the English language 0,0 only. In that body of work, we adopted the EUCs approach due to its simplicity and understandability by stakeholders, findings that were supported by our evaluations. As a result, we were keen to investigate whether the advantages we had observed with EUCs could be extended to multi-lingual requirements, namely requirements in English and Malay language

In this paper, we introduce our new Pair-Oriented Requirements Engineering (PORE) approach using the Essential Use Cases (EUCs) modeling approach to capture and analyse multi-lingual requirements. Considering students as novice practitioners (software engineers), we investigated its application with two cohorts of students, each cohort taking one of two software engineering related courses, Requirements Engineering and Software Testing, at the Universiti Teknikal Malaysia Melaka, a public university in Malaysia. Specifically, we were interested to investigate whether a pair analysis approach using EUCs leads to better results in comparison to individual analysis undertaken by students. The focus of this study is to investigate the out-come of working as a pair rather than how the students work together. Therefore, our key research question is:

"Do students working in pairs perform better than students working individually in analysing multi-lingual requirements using an Essential Use Case modelling approach?"

In this paper, we report the outcomes of the two experiments to determine the effectiveness of the PORE method for capturing and analysing multi-lingual requirements. In each experiment, the time taken and the accuracy (score) of the students' EUCs analysis of requirements expressed in both Malay and English languages are evaluated. Based on the correlation between time taken and the correctness of both languages, the results of this research indicate a positive result when PORE is used.

2 Background & Motivation

2.1 Essential Use Cases (EUCs)

The EUC approach is defined by its creators, Constantine and Lockwood, as a "structured narrative, expressed in a language of the application domain and of users, comprising a simplified, generalized, abstract, technology free and independent description of one task or interaction that is complete, meaningful, and well-defined from the point of view of users in some role or roles in relation to a system and that embodies the purpose or intentions underlying the interaction" [6]. An EUC takes the form of a dialogue between the user and the system. The aim is to support better communication between developers and stakeholders via a technology-free model and to assist better requirements capture. This is achieved by allowing only specific detail that is relevant to the intended design to be captured [7].

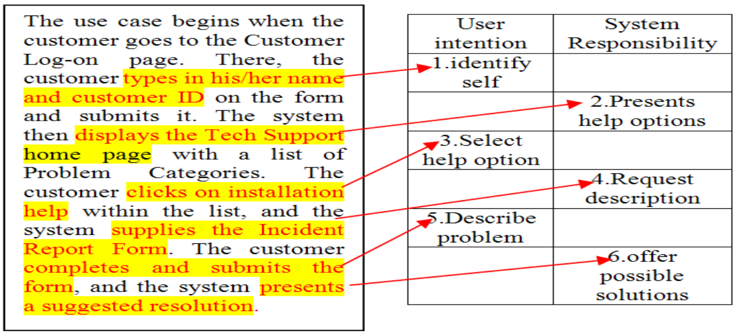

Fig. 1. Example of Natural Language Requirements and Essential Use Case model [10][8]

An EUC description is generally shorter and simpler than other requirements descriptions as it only comprises the essential steps (core requirements) of intrinsic user interest. It comprises user intentions and system responsibilities to document the user/system interaction without the need to describe a user interface in detail. The abstractions used, abstract interactions are more focused towards the steps of the use case rather than narrating the use case as a whole. The essential interactions between user and system are organised into an interaction sequence, the EUC. Figure 1 shows an example of natural language requirements (left hand side) and an EUC (right hand side) capturing these requirements (adapted from [8]). A set of essential interactions (highlighted), are extracted from the requirements. From each of these, a specific key phrase (the essential requirement of the target system it captures), called an abstract interaction, is abstracted and shown in the Essential Use case on the right as user intentions (left column) and system responsibilities (right column).

2.2 EUC Tool Support

We have developed a range of tools to support the use of EUCs. MaramaAI supports automatic extraction of EUCs from textual requirements, together with the comparison of those EUCs against the best practice EUC patterns [9] . It also supports generation of user interface models and prototypes to assist communication with stakeholders [10] and was extended to include support for multi-lingual requirements [1]. More recently, we developed a web-based tool, MEReq, which supports multi-lingual requirements capture and EUCs extraction [11] with consistency management between the multi-lingual requirements and models. For the purpose of the study presented here, the two tools described above were not used since the focus of this study is to investigate the effect of pair work on requirements capture and analysis in a multi-lingual context using EUCs.

2.3 Pair Analysis and PORE

In pair-programming, two developers sit together to work on the same code using a single computer [12],[13]. There are two roles used in pair programming: the "Driver" who types the code and the "Navigator" who observes the activity of the driver [14]. Motivated by the popularity and success of pair programming, we adapted the pairing concept to the capture and analysis of multi-lingual requirements. Essentially, the concept of driver and navigator can be used for effective requirements capture and analysis, when both users are actively communicating and discussing the task given. Thus in PORE, the same pairing approach is applied with, two roles identified: the "Codifier' who captures and analyses the requirements as EUCs and the "Navigator" who observes and checks the capture and analysis activities of the codifier.

2.4 PORE and EUCs

We considered the adoption of PORE combined with the use of EUCs for requirements analysis. Our postulation was that the accessibility of EUCs and the quality enhancements provided by pairing would result in improved quality of analyzing and capturing multi-lingual requirements, as exhibited by measures, such as the correctness of analysis and the time spent by users to complete the task.

3 Related Work

There has been much research examining pairing in the software development process that shows significant benefits to the development process and output. The most common is pair-programming. Silliti et al. [15] investigated the effects of pair-programming on developers' attention and productivity by looking at the influences of pair programming on their code writing style and their interaction with the development machine. They found that pair-programming allows the developers to stay more focussed and spend a longer time on task and switch less often between tools. However, more data is needed to support these preliminary findings.

There has been some research on pair-designing. Canfora et al. [16] implemented pair-programming in the design phase in an industry setting by having two designers work together on the same task, at the same time and on the same machine. They provided textual requirements and used the use cases and class diagram as analysis and design documents for the experiments. They found that pair-designing improved the quality but it increased the time required to complete the task [16]. Further experiments including qualitative study are in need to ensure the accuracy of the findings.

Bellini et al. [12], also conducted an experiment and its replica in both Italian and Spanish academic settings to understand the capability of pair-designing in diffusing and enforcing design knowledge when a system design is evolved. They used formalised system design documentation in UML including textual system requirements specification, use cases and class diagram. They found that pair-designing helps to increase the diffusion of the knowledge among the project team as well as providing a good level of predictability on the enforcement of knowledge compared to the traditional designing setting [12]. However, a similar experiment in industry and application of this approach to more complex systems is needed.

Albakry and Kamalrudin [5] implemented pair-analysis by adapting pair-programming to the requirements analysis process in an academic setting. They conducted a preliminary study to compare the outcomes of pair and single participants by evaluating the performance and correctness of the answers as well as the students' satisfaction and confidence [5]. Their findings were positive but require more experiments with larger groups of participants for further confirmation. Additionally, a better way to pair the students for analysis work by considering the differences of course background and culture is needed.

In summary, prior work demonstrates the benefits of pairing in software development. However, there has been a limited work on pairing in requirements engineering. No research using pairing to solve multi-lingual requirements and the Essential Use Cases as a semi-formalised way to capture requirements has been documented.

4 Research Methodology

This study was conducted to investigate the effect of pairing on the correctness and time spent of novice users working on analysing multi-lingual requirements using

EUCs. In this study, students are considered as novice users. As such, throughout this paper, the term students will be used to refer to novice users. The formulation of hypothesis, the sample, the instruments and the study procedure used in this study are described below. To address the objective of this study, a quasi-experimental study has been employed and the research design is as shown in Figure 3.

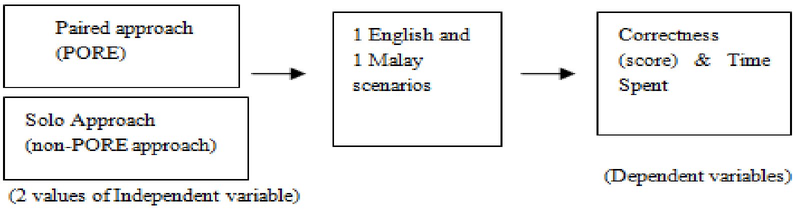

Fig. 2. Research **Design**

Using our PORE approach we had students capture and analyse multi-lingual requirements in both Malay and English languages. EUCs are used to model the captured requirements as they have proven to be useful to capture and analyse multi-lingual requirements [1]. In this respect, the effectiveness of PORE with EUCs was measured based on the correctness of the students' EUC models (score out of 6) and the time they took to capture and analyse the multi-lingual requirements. Hence, students' scores and time spent were our dependent variables, and pairing and solo approaches the independent variable. The following two hypotheses were used:

H1: There will be a significant difference in correctness between paired and solo students when analyzing multi-lingual requirements using an Essential Use Case modeling approach.

H2: There will be a significant difference in time spent between paired and solo students when analyzing multi-lingual requirements using an Essential Use Case modeling approach.

To test the reliability of our results, the quasi experiment was performed on two different cohorts. Both studies were conducted at UTeM, a Malaysian public university, but at a different time frame. The first study was done during semester II, 2011/2012, while the second was conducted during semester I, 2012/2013.

A different sample of participants was selected for the two studies: the first study involved 80 participants from a Requirements Engineering course, comprising 75 Malaysian students and 5 international students. Due to the structure of the study group, they were divided into two study sections: 40 participants per section, allowing each section to be assigned to one treatment of the independent variable. The replica study involved 38 students from a Software Testing course, comprising 36 Malaysian students and 2 international students. The structure of this study group divided the cohort into 2 sections, 28 participants in one and 10 participants in the second.

All of the participants in the two studies were proficient in both languages. The Malaysian students' level of English language proficiency was approximately equal across the cohort as all of them had achieved between Band 3 and Band 4 in the Malaysian English Language Entrance Examination (MUET), a national English language examination undertaken by Malaysian students. For the international students, they had achieved at least Band 6.5 for IELTS (International English Language Testing System) upon entrance to the university.

The instruments of the study were the two different scenarios, one in English and the other in Malay. To avoid bias, the two scenarios are for different tasks. However, the requirements are of a similar level of complexity as both requirements have an equivalent level of abstraction of the abstract interactions (see the Appendix). The level of language used for both requirements has also been verified to have similar level. The similarity in the level of complexity of the two tasks and language has been verified by an expert in requirements who is proficient in both languages.

Participants were required to capture and analyze requirements manually, using two different sets of simple requirements: one in English language and the other in Malay language. They were instructed to attempt the task manually rather than using any of the EUC support tools we had developed. This is because our main concern was to assess whether they have an understanding of capturing and analyzing requirements manually using the EUC concept, without the assistance of any tools. This is crucial, as the participants should have a strong understanding of the modeling concept in “pen and paper” rather than being dependent on any specific tool. We also wanted them to be familiar with the manual PORE approach before starting to use the supporting tools. We will explore their experiences using MEReq in a subsequent study.

4.1 Study Procedures

At the start of the academic semesters for each of the two studies, one of the authors provided the subjects with an overview of the experiment (including the pairing concept) in one of the course lectures. Prior to the conduct of the experiments, participants were given three similar exercises in both languages. This allows the participants to gain familiarity with working with EUCs and pair work.

During the experiment, two scenarios were given (refer to appendix) to the participants: reserving a vehicle (with requirements expressed in the Malay language) and Getting Cash (expressed in English). We asked them to capture these requirements in Malay and English EUCs respectively (i.e. Malay EUC model for Malay requirements and English EUC model for English requirements). The participants had to model requirements in EUCs by extracting the correct essential requirements and then model the right sequencing and responsibilities in the EUCs. The need to model both English and Malay requirements simulated the “code-switching” typical in Malay multilingual settings. Participants were given 60 minutes to complete the task.

5 Results

5.1 Study 1

A total of 80 students participated in our study. Subjects were final year Computer Science students enrolled in a Requirements Engineering course at UTeM. Participants were divided into two study groups consisting of 40 subjects per group. 40 participants attempted the tasks individually, while the other 40 participants attempted the tasks in pairs, resulting in 20 pair works. In this study, the international students were instructed to work in pairs with the Malay speakers. Table 1 shows the results of applying a bivariate Pearson correlation test to measure the association between scores and the time spent by both pair and solo students. Results show the strongest correlations between Malay and English scores ($r(60) = 0.59$, $p = .000$).

Table 1. Correlation (Scores and Time Spent) (N=60)

	EScore	ETime	MScore	MTime
EScore				
ETime	.222*	1		
MScore	.590**	.078	1	
MTime	.233*	.347**	.211	1

*. Correlation is significant at the 0.05 level (1-tailed)

**. Correlation is significant at the 0.01 level (1-tailed)

Table 2. Descriptive Statistic (Solo and Pair)

		N	Mean	SD
EScore (range: 0 to 6)	Solo	40	2.23	1.143
	Pair	20	3.10	1.252
	Total	60	2.52	1.242
ETime (range:0 to 60)	Solo	40	10.83	4.494
	Pair	20	12.95	6.083
	Total	60	11.53	5.127
MScore (range: 0 to 6)	Solo	40	2.88	.992
	Pair	20	3.55	1.276
	Total	60	3.10	1.130
MTime (range:0 to 60)	Solo	40	10.10	3.761
	Pair	20	11.30	5.686
	Total	60	10.50	4.482

Hypothesis Testing

Table 2 shows the sample size, values for mean scores and standard deviations for each group. The mean scores for the pair group are greater than the solo group. In terms of the time spent, on average solo students spent less time than paired students in both types of requirements. Hypothesis H1 and H2 were tested using an independent sample *t*-test. This test is appropriate to be used when investigating the difference between two unrelated groups on approximately normal dependent variables [23]. In our case the two groups are pair and solo, while our dependent variables are students' scores. The results from the Levene test indicate that the assumption of homogeneity of variances of each variable was not violated (i.e. F=0.142, p = 0.708 for Escore; F= 2.301, p=0.135 for ETime; F=3.266, p=0.076 for MScore; F=3.388; p=0.071 for MTime). Hence we assume that the variances of scores in the two groups are equal.

The *t*-test results (see Table 3) showed that paired groups were significantly different from solo participants on English based scores (p = 0.009). Inspection of the two group means indicates that the average score for English-based requirements for paired groups (3.10) is significantly higher than the scores for solo students (2.23). The difference between the means is 0.87 point on a 6-point test. Similarly the pair group outperformed the solo group for Malay based requirements task. Thus, based on our data we found strong support for the alternate hypothesis for H1 i.e. that there is a significant different in correctness between pair and solo students. Our results showed that the pair group performed better than the solo group.

In terms of the time spent, although the pair group in general tended to spend more time during the exercise, we could not find a statistically significant difference between the groups. The time taken by pair group in completing the exercise did not differ significantly from the solo group (p = 0.13 for English requirements; p = 0.33 for Malay requirements). Hence the null hypothesis for H2 was supported. Increasing sample size for future studies will help to increase the statistical power value, hence would give us more discrimination.

Table 3. Comparison of Pair and Solo Group on Scores and Time Spent (N_1 = 40 solo N_2=20 pairs)

	t-test for Equality of Means			
	t	df	Sig.	Mean difference
EScore	-2.71	58	0.009	-0.875
ETime	-1.53	58	0.131	-2.125
MScore	-2.26	58	0.028	-0.675
MTime	-0.98	58	0.332	-1.200

5.2 Study 2: Replication

A total of 38 students participated in the replicated study. Subjects were final year Computer Science students enrolled in a Software Testing course in Universiti Teknikal Malaysia Melaka (UTeM) and they did not participate in the earlier study. The participants had enough experience and knowledge in requirements engineering and EUC modeling specifically as they had already taken requirements engineering and software engineering subjects before. As in the first study, the research is organised in two Sections of the course. Twenty eight (28) participants in Section 1 were required to solve the task in solo, while 10 participants in Section 2 worked in pairs, resulting in 5 paired works. In this study, international students were instructed to work in pairs since Malay is not their primary language.

Table 4 shows results from applying a bivariate Pearson correlation test to measure the association between scores and the time spent by both pair and solo students. Results show positive and significant correlation between the time spent for the Malay and English requirements. ($r(38) = 0.32$, $p = .001$).

Hypothesis Testing

Table 5 shows the sample size, values for mean scores and standard deviations for each group. Our data shows that mean scores for pair groups are greater than the solo groups for both Malay and English requirements. In terms of time spent, on average solo groups spent less time than paired students for Malay requirements. However, for English requirements, the paired groups spent less time than solo students. We think that this is due to a few subjects in the paired group being international students. They needed more time for discussion with their partner who was a native Malay speaker.

Table 4. Correlations between Scores and Time Spent (N=38)

	EScore	ETime	MScore	MTime
EScore	1			
ETime	.174	1		
MScore	-.093	.133	1	
MTime	.147	.323*	.089	1

*. Correlation is significant at the 0.05 level (2-tailed)

Table 5. Descriptive Statistic (Solo and Pair)

		N	Mean	SD
EScore (range: 0 to 6)	Solo	28	1.64	0.911
	Pair	10	1.80	0.919
	Total	38	1.68	0.904
ETime (range:0 to 60)	Solo	28	14.07	5.741
	Pair	10	13.60	4.575
	Total	38	13.95	5.402
MScore (range: 0 to 6)	Solo	28	2.50	0.577
	Pair	10	3.60	0.843
	Total	38	2.79	0.811
MTime (range:0 to 60)	Solo	28	14.89	5.705
	Pair	10	18.50	11.816
	Total	38	15.84	7.765

Table 6. Comparison of Pair and Solo Group on Scores and Time Spent (N1= 28 solo, N2=10 pairs)

	t-test for Equality of Means			
	t	df	Sig.	Mean difference
EScore	-.467	36	0.643	-0.157
ETime	-.260	36	0.816	0.471
MScore	-1.747	36	0.000	-1.100
MTime	-1.271	36	0.212	-3.607

The hypothesis was also tested using an independent sample t-test and the Levene test. The Levene test indicates that the assumption of homogeneity of variances of each variable was not violated (i.e. F=0.38, p = 0.847 for Escore; F= 0.26, p=0.613 for ETime; F=2.015, p=0.164 for MScore; F=1.423; p=0.241 for MTime). Hence we assume that the variances of scores in two groups are equal.

The t-test results (see Table 6) showed that there were no significant differences in correctness between paired students and solo students (ρ = 0.643) for English requirements. Similarly, there were no significant differences on the time taken to solve the requirements in English (ρ =0.816). However, for the Malay requirements, we found that there is a significant difference in correctness between paired and solo students (ρ = 0.00), thus supporting our hypothesis. However, we could not find a statistically significant difference between paired and solo students in terms of the time spent to solve a task written in Malay. Based on these results, we found evidence that pairing work has benefited students when analysing Malay requirements. This was consistent with the results from our previous study.

6 Discussions

Based on the results presented in the previous section, we found evidence that PORE is able to help novice users perform better in requirements engineering in captured requirements quality score. This was particularly useful when analysing requirements written in the Malay language. However, our results showed only partial support for users working in pairs on English requirements having improved captured requirements quality scores. In terms of the time spent, we found that there was no evidence to differentiate the time taken to solve the task between pair and solo group for both languages. The aggregation of the hypothesis testing results of each study is presented in Table 7.

Table 7. Summary of Findings

Study	**Supported Hypothesis - pair group outperform solo? (YES/NO)**			
	H1: Correctness (scores)		**H2: Time Spent**	
	Malay	English	Malay	English
Study 1	YES	YES	NO	NO
Study 2	YES	NO	NO	NO

There are some uncontrolled variables that may have affected the validity of these experimental results. One of these was the language that is the mother tongue of our subjects. In this study, most of our subjects were Malay native speakers and only a few were international students who came from other countries, such as China and the Middle East. The ability of these students to comprehend and analyse requirements written in Malay may be limited as compared to requirements written in English. Other uncontrollable variable that may affect the findings of this study are the level of knowledge of each individual, their cultural background and personality.

There is also a possibility that the level of difficulty or complexity of the task may have influenced the results and students may have spent more time working on the more difficult requirements set. We believe that further work is needed in order to investigate the impact that task complexity has upon PORE's effectiveness.In terms of the time taken to analyse English requirements, we found from both studies that the pair group spent a little longer than the solo group; however the results were not statistically significant. We speculate that there is be a greater amount of communication and discussion among paired group when compared to solo group but that this discussion leads to a solution as quickly as with individuals and with some evidence that this is typically a better solution. This is because two heads can have different understanding, thus they might suggest different ideas and solutions exploring the solution space more efficiently. Similar findings appear for both groups of students working on Malay requirements. We suggest future work should include a larger sample size to confirm or refute current findings.

7 Conclusions and Future Work

We have described our newly developed requirements engineering method called PORE. PORE is used together with the Essential Use Case model (EUC) to analyse multi-lingual requirements (i.e. those using English and Malay language). In this paper, we presented a study and its replication analysing a set of multi-lingual requirements using undergraduate students as subjects. The results obtained partially support our proposition that novice users exhibit better performance in term of correctness and time spent in analyzing multi-lingual requirements when working in pairs. We found that in both studies, the pair group outperformed the solo group for the Malay-based requirements task. For the English-based requirements, we found such supporting evidence in Study 1 only. Our results showed that there was no significant difference in terms of the time spent to analyze the tasks between pair and solo groups. We speculate that task complexity might play a role in influencing this result. We also anticipate that the study will give different findings if users are asked to extract an EUC model in a different language from the provided requirements language.

For future work, we plan to conduct more replications of our study with a larger number of students in Requirements Engineering and Software Engineering. We also plan to explore this approach with other modeling languages such as UML use case, sequence and class diagrams and then compare them with our findings. We also plan to use other languages to support our proposition that PORE is able to improve the quality of requirements models in analysing requirements in a multi-lingual context. We also intend to explore PORE usage in industry to identify the benefits of implementing this method in analysing requirements in real business activity. However, practitioners are more experienced as they have had professional training. Hence, we will use more complex requirements for this study. Finally, we plan to embark on a PORE study using our developed support tools, such as MEReq [1], contrasting it with the study reported here which has focused on a paper-based capture and analysis of multi-lingual requirements.

8 Acknowledgments

This research is funded by the Ministry of Higher Education Malaysia (MOHE), Universiti Teknikal Malaysia Melaka (UTeM) and Swinburne University of Technology. We also would like to acknowledge Pn. Nor Haslinda Ismail for allowing us to conduct the experiments in her class.

9 References

1. Kamalrudin, M., Grundy, J., & Hosking, J. Supporting requirements modelling in the Malay language using essential use cases. In*Visual Languages and Human-Centric Computing (VL/HCC), 2012 IEEE Symposium on* (pp. 153-156). IEEE (2012)
2. Regev, G., & Wegmann, A. Where do goals come from: the underlying principles of goal-oriented requirements engineering. In *Requirements Engineering, 2005. Proceedings. 13th IEEE International Conference on* (pp. 353-362). IEEE (2005)
3. Goedicke, M., & Herrmann, T. A case for viewpoints and documents. In *Innovations for Requirement Analysis. From Stakeholders' Needs to Formal Designs* (pp. 62-84). Springer Berlin Heidelberg (2008)
4. Chen, Z., & Ghose, A. Web agents for requirements consistency management. In *Web Intelligence, 2003. WI 2003. Proceedings. IEEE/WIC International Conference on* (pp. 710-713). IEEE (2003)
5. Albakry, K., & Kamalrudin, M. Pair analysis of requirements in software engineering education. In *Software Engineering (MySEC), 2011 5th Malaysian Conference in* (pp. 43-47). IEEE (2011)
6. Constantine, L. L., & Lockwood, L. A. (1999). *Software for use: a practical guide to the models and methods of usage-centered design*. Pearson Education (1999)
7. Biddle, R., Noble, J., & Tempero, E. Essential use cases and responsibility in object-oriented development. In Australian Computer Science Communications (Vol. 24, No. 1, pp. 7-16). Australian Computer Society, Inc. Chicago (2002)
8. Constantine, L. L., & Lockwood, L. A. Structure and style in use cases for user interface design. Object modeling and user interface design, 245-280 (2001)
9. Kamalrudin, M., Hosking, J., & Grundy, J. Improving requirements quality using essential use case interaction patterns. In Proceedings of the 33rd International Conference on Software Engineering (pp. 531-540). ACM (2011)
10. Kamalrudin, M., & Grundy, J. Generating essential user interface prototypes to validate requirements. In *Proceedings of the 2011 26th IEEE/ACM International Confer-*

ence on Automated Software Engineering (pp. 564-567). IEEE Computer Society (2011)

11. Kamalrudin, M., Grundy, J., & Hosking. *MaramaAI: Tool support for capturing requirement and checking the inconsistency* in *21st Australian Software Engineering Conference*. 2010. Auckland, New Zealand: IEEE Computer society (2010)
12. Bellini, E., Canfora, G., García, F., Piattini, M., & Visaggio, C. A. (2005). Pair designing as practice for enforcing and diffusing design knowledge. *Journal of* Software Maintenance and Evolution: Research and Practice, 17(6), 401-423 (2005)
13. Braught, G., Eby, L. M., & Wahls, T. The effects of pair-programming on individual programming skill. In ACM SIGCSE Bulletin (Vol. 40, No. 1, pp. 200-204). ACM (2008)
14. Gehringer, E. F. A pair-programming experiment in a non-programming course. In *Companion of the 18th annual ACM SIGPLAN conference on Object-oriented programming, systems, languages, and applications* (pp. 187-190). ACM (2003)
15. Sillitti, A., Succi, G., & Vlasenko, J. (2012, June). Understanding the impact of pair programming on developers attention: a case study on a large industrial experimentation. In *Software Engineering (ICSE), 2012 34th International Conference on* (pp. 1094-1101). IEEE (2012)
16. Canfora, G., Cimitile, A., Garcia, F., Piattini, M., & Visaggio, C. A. (2007). Evaluating performances of pair designing in industry. *Journal of Systems and Software*, *80*(8), 1317-1327 (2007)

10 Appendix: Requirements Used

Malay requirements:

1. "*Use Case*" bermula apabila pengguna menyatakan hasrat utk membuat tempahan untuk menyewa kereta.
2. Sistem bertanyakan tempat untuk mengambil dan menghantar tempahan besertakan tarikh dan masa untuk mengambil tempahan. Pengguna menyatakan tempat dan tarikh yang dikehendaki.
3. Sistem bertanyakan jenis kenderaan yang dikehendaki oleh pengguna. Pengguna menyatakan jenis kenderaan yang dikehendaki.
4. Sistem memaparkan semua kenderaan yang sesuai dengan tempat untuk mengambil kenderaan berdasarkan tarikh dan masa yang dikehendaki. Sekiranya, pengguna mengkehendaki maklumat lanjut tentang kenderaan yang spesifik, sistem memaparkan maklumat tersebut kepada pengguna.
5. Sekiranya pengguna memilih sebuah kenderaan untuk tempahan, sistem pun meminta maklumat untuk mengenalpasti pengguna (nama penuh, nombor telefon, alamat emel, alamat untuk kenalpasti,dll). Pengguna memberi maklumat yang dikehendaki.
6. Sistem memaparkan maklumat untuk keselamatan (cthnya,perlindungan kebinassan,insuran kemalangan peribadi) dan bertanyakan samada pengguna menerima atau menolak setiap produk. Pengguna menyatakan pilihannya.
7. Sekiranya, pengguna menyatakan hasratnya untuk "menerima tempahan", sistem akan memberitahu pengguna bahawa tempahannya sudah selesai dan memaparkan

pengesahan tempahan kepada pengguna. "*Use case*" ini tamat apabila pengesahan tempahan telah ditunjukan kepada pengguna.

User Intention	System Responsibility
1. membuat pilihan	
	2. memberi pilihan
	2. memaparkan maklumat
	4. meminta pengesahan
5. memberi maklumat	
	6. mengesahkan tempahan

Fig. 3. The EUC Requirements model in the Malay language

English requirements:

1. The use case begin when the Client insert an ATM card. The system reads and validates the information on the card.
2. System prompts for pin. The client enters PIN. The system validates the PIN.
3. System asks which operation the client wishes to perform. Client selects "Cash withdrawal."
4. System request amounts. Client enters amount.
5. System request type. Client selects account type (checking, saving, credits)
6. The system communicates with the ATM network to validate account ID, PIN and availability of the amount requested.
7. The system asks the client whether he or she wants receipt. This step is performed only if there is paper left to print the receipt.
8. System asks the client to withdraw the card. Client withdraws card. (This is security measure to ensure that clients do not leave their cards in the machine.)
9. System dispenses the requested amount of cash.
10. System prints receipt.
11. Client receives cash
12. The use case ends.

User Intention	System Responsibility
1. Identify self	
	2.verify identity
	3.offer choices
4.choose	
	5.Dispense Cash
6. Take Cash	

Fig. **4.** The EUC Requirements model in English language