MEReq: A Tool to Capture and Validate Multi-Lingual Requirements

Massila Kamalrudin,Safiah Sidek and NoorrezamYusop [a] , John Grundy [b] and John Hosking[c]
[a] *Innovative Software System and Services Group, Universiti Teknikal Malaysia Melaka,Melaka Malaysia*
[b] *Center for Computering and Engineering Software Systems, Swinburne University of Technology, Victoria, Australia*
[c] *Center of Engineering & Computer Science, Australia National University, Canberra, Australia*

Abstract. Within the era of globalisation that acknowledges differences and diversity, multiple languages have been increasingly used to capture requirements. This practice is particularly prevalent in Malaysia, where both Malay and English languages are used as a media of communication. Nevertheless, capturing requirements in multiple languages is often error-prone due to natural language imprecision being compounded by language differences. Considering that two languages may be used to describe requirements for the same system in different ways, we were motivated to develop MEReq, a tool which uses Essential Use Case (EUC) models to support capturing and checking the inconsistency occurring in English and Malay multi-lingual requirements. MEReq is tablet compatible to minimise time for on-site capture and validation of multi-lingual requirements. This paper describes the MEReq approach and demonstrates its use to capture and validate English and Malay requirements.

Keywords. Multi-lingual requirements, Requirements capture, Consistency checking, Essential Use Cases model

Introduction

Requirements are typically identified during the early phase of system development and need to conform to the needs and expectations of the system's clients [1]. A complete software requirement specification accepted by developers and clients is the result of a shared understanding and agreement of what and why a software system should do.

In the software industry, English has been recognised as the defacto common natural language used to write and describe requirements. However, in the current era of globalisation, which acknowledges differences and diversity, different languages are beginning to be used to elaborate requirements. Hence, there is a need to support multilingual requirements capture, particularly in multi-lingual societies.

This research involves the use of multiple languages, namely English and Malay, to capture and validate requirements within the context of the software industry in Malaysia. Malaysia is a multi-cultural, multi-lingual country. Bahasa Malaysia, or the Malay language, is the official language, while English is the second language [2]. As

an official language of Malaysia, most official communications are written in Malay [2]. However, English is still a preferred working language, especially in the private sector. In software development organisations, "code-switching" between Malay and English languages is a common practice [2].

This motivated us to explore support for multi-lingual requirements capture and validation, as two languages may be used to describe requirements for the same system in different ways introducing new possibilities for inconsistency [6]. This is related to our previous work, where we developed a technique and toolset supporting English requirements capture and consistency management. Our extracted EUC models were used to perform a more detailed analysis, enabling the identification of a range of potential problems with the extracted requirements, such as inconsistency, incompleteness and incorrectness [5, 8].

Here we present our new work on a refined approach to support the process of capturing and checking the consistency of multi-lingual requirements in both English and Malay languages. This approach, MEReq, uses Essential Use Case (EUC) models derived from English and Malay language interaction pattern libraries.

1. Motivation and Related Work

Multi-lingual requirements are commonly used in countries where English is not their native language and in outsourcing situations for Global Software Development. Several research projects have identified that this leads to challenges, especially requirements inconsistency and misinterpretation, due to factors such as different cultural and language usage [12], [13],[5]. Calefato et al. [4] explored automated cross language translation to overcome the barrier of communication in requirements engineering. Their work focused on English and Italian and on the quality of the translation rather than consistency issues between and within the requirements.

Some research has explored handling natural language (NL) specifications written in Malay. For example Shukur et al. [3] employed formal approaches for translating a software specification in Malay into formal Z statements using their tool, M2Z. The tool is limited to certain grammar, rules and data types and only works for small types of statements [3]. Ab Aziz et al. [14] developed a new technique for extracting Malay grammar by introducing the pola grammar technique using automata and finite states to investigate the relationship between parsing and corpus method in Malay [14]. Much research focuses on capture and consistency validation of NL requirements. Holtmann et al. validated consistency and completeness of system requirements in the automotive domain [15]. They proposed a formal mechanism with controlled NL to document requirements and use procedure and requirements patterns to check for inconsistency, incompleteness and other problems. This work also supports REs with corrective action to amend requirements when faults are found. Kamalrudin et al. [9],[5] developed a technique and toolset, MaramaAI to support requirements capture and consistency management using EUCs. This tool is supplemented by end-to-end rapid prototyping support. The tool uses EUC patterns to validate requirements consistency, completeness and correctness. However, it focuses on English requirements only. This tool was extended to support capture and consistency management of Malay requirements by adapting the end-to-end rapid prototyping framework [11],[17]. Although this tool supports requirements capture and management in both English and

Malay simultaneously, it does not have support for checking consistency between the requirements in each of the languages.

In summary, much research has been devoted to handling NL requirements and improving their consistency. Although inconsistency is a particularly common issue in multi-lingual requirements, there is little work focussed on capturing and managing consistency of multi-lingual requirements, especially between English and Malay.

2. Approach and Prototype Tool

Our new toolset, MEReq (Malay-English Requirements), is designed to overcome the problems of capturing and checking consistency of multi-lingual requirements. Like our previous work, this adopts an EUC-based approach. Figure 1 shows an EUC model extracted from textual requirements at left. A set of "essential interactions", phrases that occur in the NL text (highlighted at left), are mapped to one or more "abstract interactions" that form the EUC dialogue, shown on the right. This abstracts the original NL requirements into a more technology-free, user system dialogue. In our previous work we supported extraction of EUCs from English [5, 7, 9] and parallel extraction from English and Malay requirements [11], but without cross-language checking.

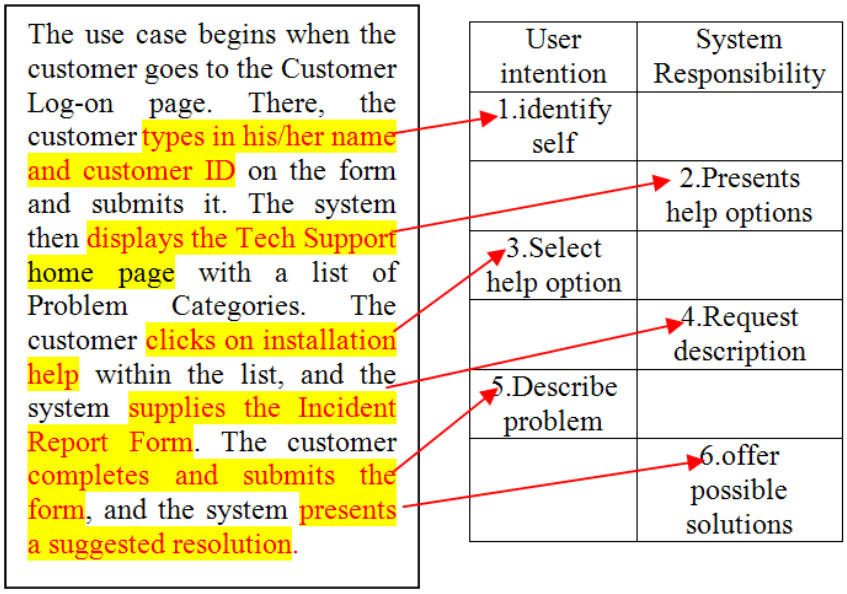


Figure 1. Example of generated EUC model (right) from the textual natural language requirements (left) adapted from [20]

Table 1 shows examples of essential interactions and their corresponding abstract interaction both in English and Malay. Abstract interaction patterns for both languages are associated with more than one essential interaction for various application domains. Abstract interactions are associated with multiple domains of IT application such as online business, e-commerce and online reservation.

Figure 2 outlines our MEReq approach that supports multi-lingual requirements engineering with EUCs. As shown in Figure 2, a new extraction engine (2) uses an essential interaction patterns library to map phrases (the essential interactions) to a list of abstract interactions. This list is then used to generate an initial EUC model in Malay or English. These models can be further refined by the RE and checked against the best-practice EUC patterns (developed by reusing our previous approach in [5]) and its proven enhancement of quality (4). Then, both generated EUC models can be compared to check for consistency between the different language models (3). Inconsistency or missing elements in the NL requirements are highlighted.

Table 1. Example English and Malay Essential Interactions

English Essential Interaction Patterns Library		Malay Essential Interaction Patterns Library	
Essential Interaction	Abstract Interaction	Essential Interaction	Abstract Interaction
1. Save record 2. Save information 3. Save data	Save information	1. Menyimpan data *(save data)* 2. Menyimpanmaklumatperibadi*(save personal information)* 3. Menyimpanrekodjualan *(save sales record)*	SimpanMaklumat*(save information)*

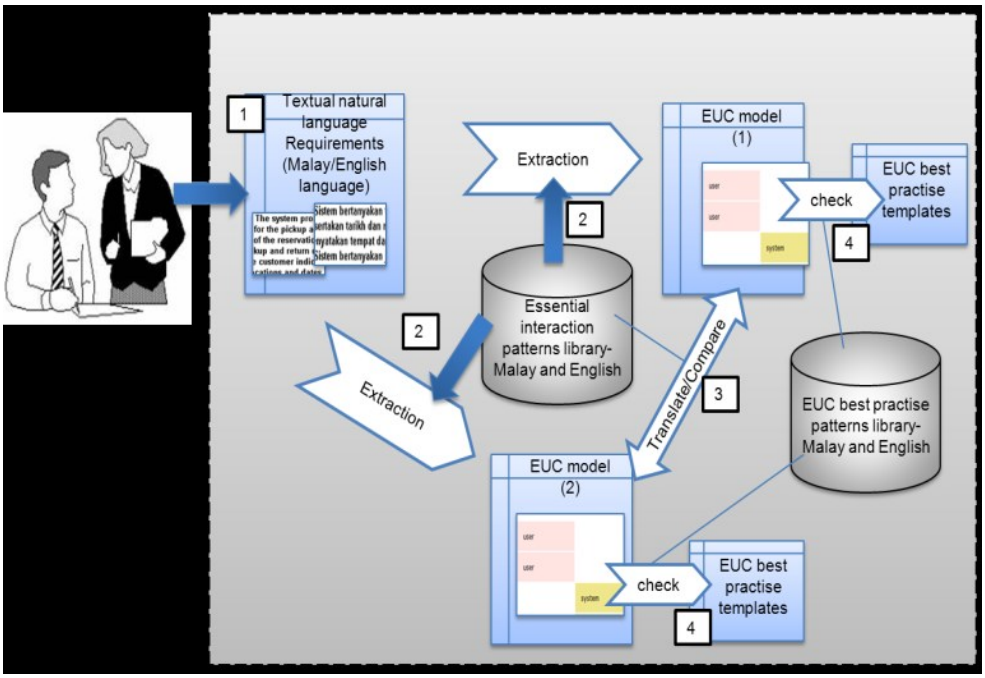


Figure 2. Overview of MEReq Approach.

Our MEReq[1] tool also uses a more accessible platform of web and mobile-based (iPad) interfaces than do our previous Eclipse-based toolsets. Using MEReq, the English and Malay textual natural language requirements are automatically extracted and visualised as EUC models. The requirements engineer can analyse the interactions and the essential requirements of both language models of the requirements at the same time. Then, consistency checking of both models can be done by using the compare and translate components.

3. Usage Example

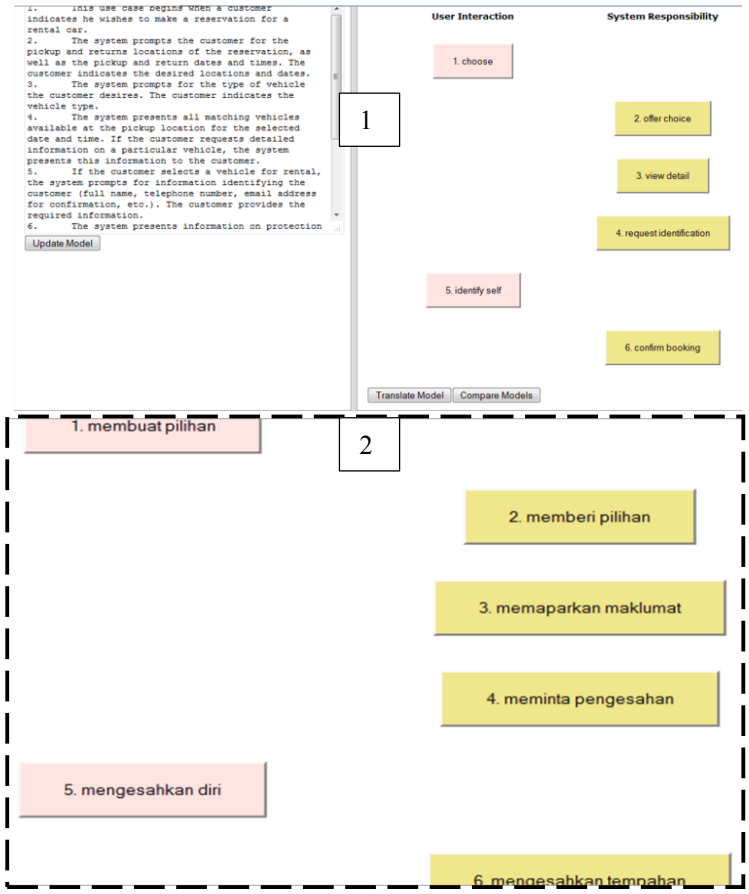


Figure 3. Translating English Language Reqirements (1) in an extracted EUC to a Malay Language EUC (2).

[1] http://www.mereq.com

We use an example scenario of reserving a vehicle *(PenempahanKenderaan)* to illustrate multi-lingual requirements capture and consistency checking using MEReq. Figure 3(1) shows some English language requirements and their extracted EUC. Figure 3(2) shows the translation of that EUC to Malay. MEReq maps the abstract interactions and interaction sequences from one language model to the other, taking into account the differences in the number and sequence of the abstract interactions in some situations due to differences in expressing the same concept in each underlying NL. This translation eases the burden on the requirements engineer to communicate with stakeholders who usually have better understanding of either one of the languages in use.

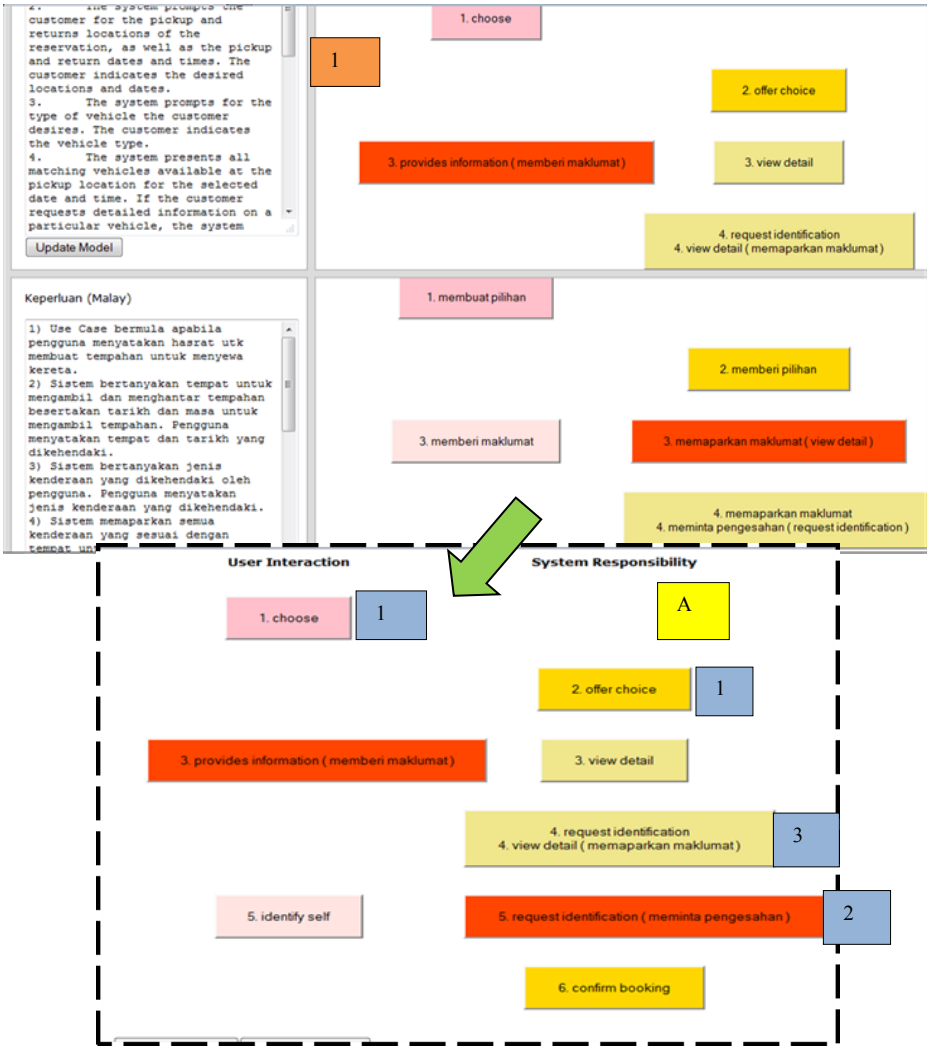


Figure 4. Capturing multi lingual (English (1A) and Malay (1B)) Languages Requirements (1) in EUC; Compare Requirements: English language EUC with Malay language EUC (2)

Figure 4(1) shows consistency checking between English and a Malay EUC models. MEReq compares the two EUCs' abstract interactions to determine whether the elements have differently-named or sequenced elements, as well as elements existing in one but not the other. To do this MEReq uses the translation mapping mechanism show in Figure 3, combined with a model comparison mechanism. The visualization (1A) shows the potential inconsistencies existing between the EUCs. The pink and yellow elements (A1) show abstract interactions that appear consistent. Here, the first two abstract interaction components "choose" *(membuatpilihan)* and "offer choice" *(memberipilihan)* are consistent in both models.

However, after that the two models diverge in sequence and structure. The red elements (A2) indicate elements occurring in one sequence but not the other, indicating inconsistency, while the mustard elements indicate inconsistent naming of elements in either sequence (A3). Both indicate fundamental disagreement in the different source natural language textual requirements that must be corrected.

4. Evaluation and Results

We conducted three studies to evaluate the efficacy and effectiveness of our approach and MEReq tool support. Firstly, we evaluated IT professional and IT student performance *manually* extracting EUCs and making these consistent without MEReq. Then we evaluated the usability of MEReeq for performing these tasks. Finally, we evaluated how well MEReq supports automated extraction of multi-lingual EUCs.

4.1. Capture of Multi-lingual Requirements

We conducted two separate quasi-experiments involving 13 Malaysian software professionals and 40 Malaysian undergraduate students. The experiments compared their performance in terms of correctness (qualitative score) and the time taken (minutes) when *manually* extracting multi-lingual requirements (in both Malay and English) to EUCs.

Part 1 Evaluation (Industry)

Participants were IT professionals currently working who had roles such as a system analyst, software engineer, IT manager or business analyst. Their work experience varied from 2 to 5 years, the majority being more than 2 years. Subjects were given a set of scenarios on *reserving a vehicle.* We asked them to capture these requirements in both Malay and English. Hence, the same scenario was given in both languages. The maximum possible score for a scenario was six, determined by evaluating resulting EUC completeness, correctness and consistency. We measured both their scores and time taken.

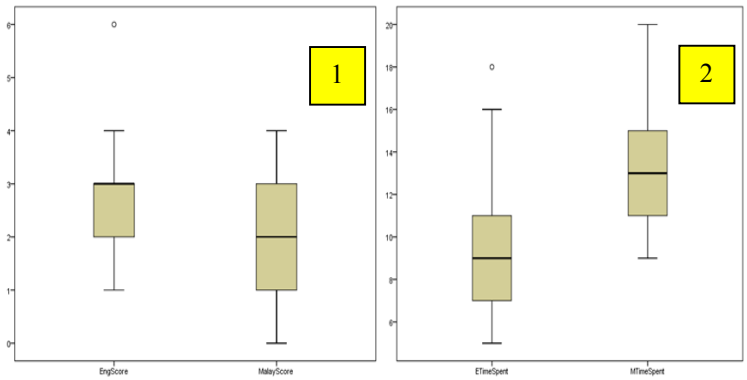


Figure 5. Distribution of Scores (1) and Distribution of time spent (2) (Industry data)

Table 1. Paired Samples statistics (N=13)

Variables		Mean	Median	Std. Dev.
Pair 1	EngScore	2.77	3.0	1.24
	MalayScore	2.08	2.0	1.26
Pair 2	ETimeSpent	9.92	9.0	3.97
	MTimeSpent	13.00	13.0	3.06

Figure 5 and Table 2 show distributions of scores and time. A median value line is shown. The boxplot shows the median score for English requirements is considerably higher than for Malay. There is an outlier for English, representing those who obtained a full mark. From the descriptive data (Table 2), the mean score for English was 2.77 out of 6 and average time was 9.92 minutes. The mean score for Malay requirements was slightly worse at 2.08 out of 6 and the time spent for this was also somewhat greater with an average of 13 minutes. The total correctness for English requirements was 36 out of 78 with the ratio of 0.46 and for Malay requirements; the total correctness was 36 out of 78 with the ratio of 0.35.

A Paired-Samples T-Test was conducted to compare whether there was any significant different between the means of the two sets of variables. Table 2 shows two pairs of data on “scores” and “time”. The paired samples test results showed that there was a significant difference in the time spent for English (M=9.92, SD=3.96) and Malay requirements (M=13.0, SD=3.05); t(12) = -3.36, p = 0.006 using the 95% confidence interval (see Table 3). This suggests that local IT professionals spend a significantly longer time to capture Malay requirements compared to English requirements. However, the results showed that there is no significant different in scores of English and Malay requirements.

Table 2. Paired Samples Test (N=13)

Variables		Paired Differences		t	df	Sig. (2-tailed)
		Mean	SD			
Pair 1	EngScore - MalayScore	0.69	1.79	1.39	12	0.190
Pair 2	ETimeSpent - MTimeSpent	-3.08	3.30	3.36	12	0.006

Part 2 Evaluation (Students)

We wanted a larger sample of requirements engineers to study. Subjects involved in the second study were 40 undergraduate students enrolled in a Requirements Engineering course at Universiti Teknikal Malaysia Melaka. All subjects were local Malay students. 40% were male, 60% female. Figure 6 and Table 4 show the resulting distributions. The scores show a broader distribution for the English scores. The outlier above the third quartile represents cases where students obtained high scores for capturing Malay requirements (i.e. scored 5 out of 6). The median scores indicate students scored higher marks for Malay requirements compared to English requirements. The time distributions show the median value of time taken for Malay and English requirements to be very similar, the broader distribution being time spent for English requirements. Students spent up to 25 minutes when dealing with English requirements compared to only only 15 minutes for Malay requirements.

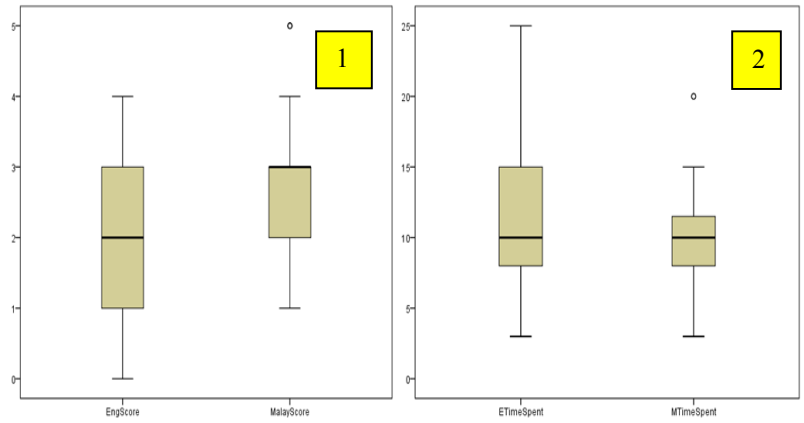


Figure 6. Distribution of Scores (1) and Distribution of time spent (2) (Student data)

The total correctness for English requirements was 89 out of 240 with the ratio of 0.37 and for Malay requirements; the total correctness was 115 out of 240 with the ratio of 0.48. We again used a Paired-Samples T-Test to analyze whether there was any significant difference in the means of two variables. This showed a significant difference in the scores obtained for English requirements (M=2.23, SD=1.14) and Malay requirements (M=2.88, SD=0.99); t(39) = -3.74, $p = 0.001$ at a 95% confidence interval (see Table 5). This shows student performance is better when dealing with

Malay requirements compared to English. There was no significant difference in time taken when capturing Malay or English requirements.

Table 3. Mean, median and standard deviation of scores and time spent for capturing requirements (N=40)

Variables		Mean	Median	Std. Dev.
Pair 1	EngScore	2.23	2.00	1.14
	MalayScore	2.88	3.00	0.99
Pair 2	ETimeSpent	10.83	10.00	4.49
	MTimeSpent	10.10	10.00	3.76

Table 4. Paired Samples Test (N=40)

		Paired Differences		T	df	Sig. (2-tailed)
		Mean	SD			
Pair 1	EngScore - MalayScore	-0.65	1.09	-3.74	39	0.001
Pair 2	ETimeSpent -MTimeSpent	0.72	4.94	0.93	39	0.359

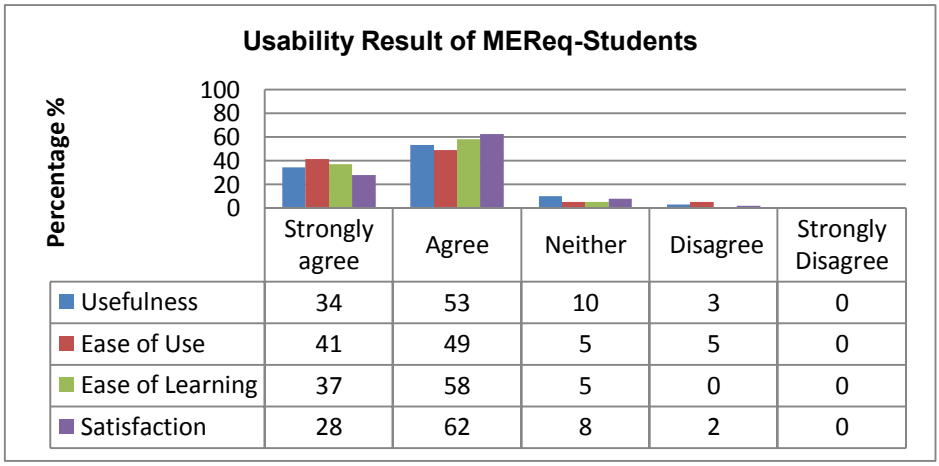
In summary, experienced practitioners took less time capturing English requirements using EUCs and tended to produce better results than when using Malay. This may be because they are more familiar with English written requirements. In contrast, students significantly increased performance working in Malay. Students are all native Malay speakers so they have seldom dealt with English requirements. As they are learning RE they are equipped with only basic RE techniques. In contrast our practitioners deal more with English requirements than Malay, even though Malay is their mother-tongue. Thus they may be more familiar with English requirements compared to Malay requirements.

We cannot make a direct comparison between students and practitioners due to the different level in background and skills. However, the overall results show end users find manual extraction of EUCs difficult, time consuming and error prone. Overall, performance results by both subject sets for both languages are poor with a correctness ratio of 37%-46% for English and 35%-48% for Malay. Even with the basic scenarios used, it is time consuming to capture manually multi-lingual requirements in EUCs. Average time was between 9-11 minutes for English requirements and 10-13 minutes for Malay requirements. These results are not atypical and are similar to our previous study [7].

4.2. Usability Evaluation of MEReq

We then asked all the IT practitioners and RE students to use our tool MEReq. All participants were given initial training. They inserted the same requirements as the manual study into the tool and then generated EUCs for both languages. They also explored the tool's facilities to check for consistency between both language requirements and EUCs. Having familiarized themselves with the tool's capabilities, they carried out another multi-lingual requirements exercise including consistency analysis. They then completed a three part survey comprising: (1) a standard evaluation of usability based on user perceptions of usefulness, ease of use, ease of learning and satisfaction from Lund [19]; (2) a set of questions to determine user perceived strengths based on selected dimensions of the cognitive dimensions of notations (CD) [20], a common approach for evaluating visual language environments, the questions being adapted from [21]; and (3) open ended questions related to improvements participants desired. For (1) and (2) a five part Likert scale: 1=strongly disagree, 2=disagree, 3= undecided, 4= agree and 5=strongly agree was used. We also observed participants' performance using the tool to accomplish the required tasks. Participants were asked to think aloud and provide suggestions to enhance the tool.

Figures 7 and 8 show usability results. For each characteristic, the results of each corresponding question block were averaged to produce the results shown. Results are very positive, with strong agreement over the usefulness of the tool (90% of practitioners and 87% of students strongly agree or agree), ease of use (87% and 90%), ease of learning (95% and 85%) and satisfaction (over 90% for both). The small numbers of cases of disagreement are related to the constraints of our Malay essential interaction pattern library. Some participants also have a preference for an indicator/legend for the type of colour used and want better highlighting in NL requirements when tracing back. Incomplete inconsistency notification or feedback on the identified inconsistencies was also reported.



	Strongly agree	Agree	Neither	Disagree	Strongly Disagree
Usefulness	34	53	10	3	0
Ease of Use	41	49	5	5	0
Ease of Learning	37	58	5	0	0
Satisfaction	28	62	8	2	0

Figure 7. Usability results- RE Students

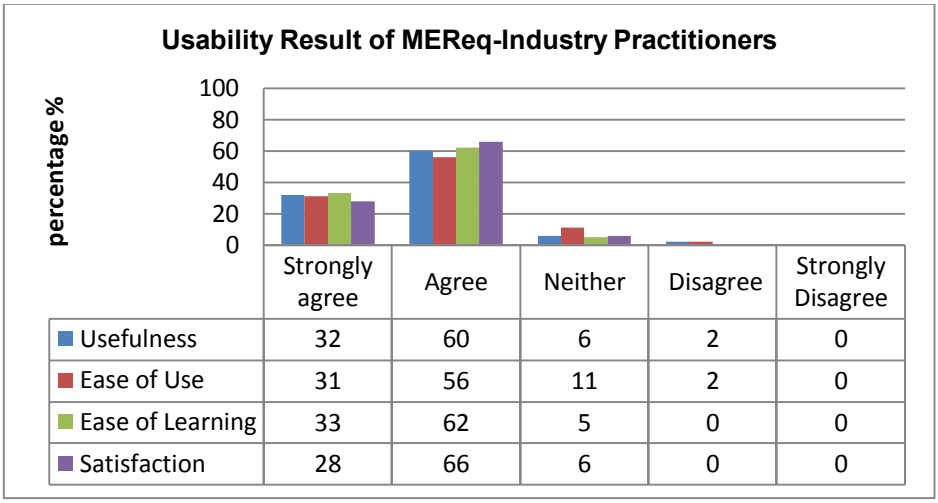


Figure 8. Usability Results- IT Practitioners

Table 6. CD Notations Used and Questions Evaluating Them

Cognitive Dimension	Question
Visibility	It is easy to see various parts of the tool
Viscosity	It is easy to make changes
Diffuseness	The notation is succinct and not long-winded
Hard mental effort	Some things do require hard mental effort
Error-proneness	It is easy to make errors or mistakes
Closeness of mapping	The notation is closely related to the result
Consistency	It is easy to tell what each part is for when reading the notation
Hidden dependencies	The dependencies are visible
Progressive evaluation	It is easy to stop and check my work so far
Premature commitment	I can work in any order I like when working with the notation

The CD study allowed us to explore in more detail reasons for user perceptions. Table 6 shows each dimension we evaluated and the question addressing it. Figures 9 and 10 show the results for both sets of participants. We found interesting trade-offs between dimensions we feel have contributed to the strong usability acceptance. The strong *viscosity, visibility and closeness of mapping* rating and the relatively high disagreement on the *hard mental effort* and *error proneness* ratings point to the EUC notations used being seen as relatively intuitive and understandable. This is in strong contrast to the difficulty found by users in the manual studies. The automation of capturing multi-lingual requirements with EUCs and automated consistency checking support appears to be responsible for this, as demonstrated by the high ratings for visibility of dependencies, consistency of notations, progressive evaluation and lack of premature commitment.

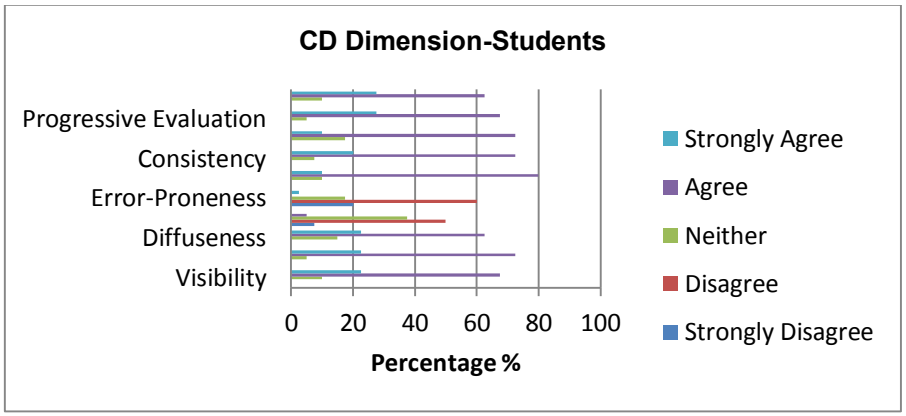


Figure 9. Cognitive Dimension Results-Students

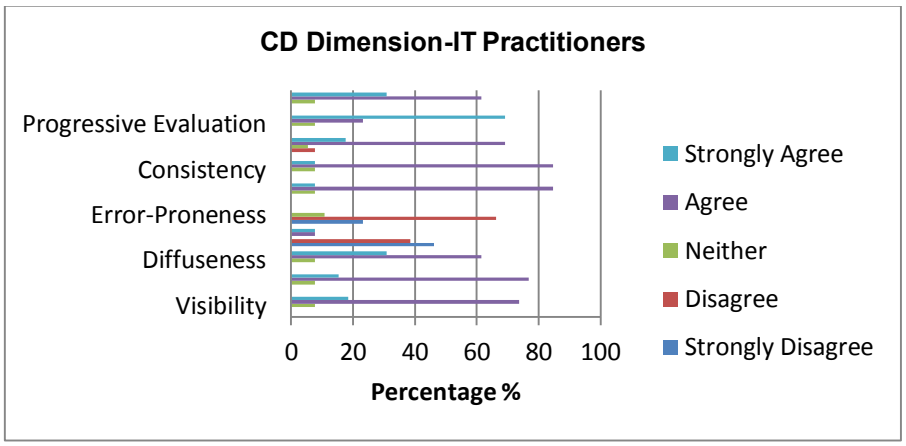


Figure 10. Cognitive Dimension Results-IT Practitioners

5. Conclusions

MEReq aims to enhance correctness and reduce time when capturing multilingual requirements. MEReq takes into account abstract interaction and essential interaction sequence translation between languages, hence helps to ensure both correctness of translated EUCs and the expression of the EUC in each language. Users of our MEReq prototype tool have been positive about its features and usability. They have successfully used it to carry out some representative multi-lingual requirements engineering capture and validation tasks.

This initial end-user experience has identified several areas for further research. A larger library of Malay essential interaction and EUC patterns needs to be developed for other target application domains. We plan to enhance the extraction engine using an intelligent algorithm and glossary to improve the scalability of the extraction process on the essential interactions and abstract interactions. We also want to continue to generalise the tool to support other languages, such as Mandarin. Further enhancement

to support extracted EUC quality checking is needed too as well as a comprehensive end user evaluation.

6. Acknowledgements

We are grateful for support from a research grant of Fulgent Corp, USA, MBIE Software Process and Product Improvement project grant, UniversitiTeknikal Malaysia Melaka (UTeM), ARC Linkage grant LP130100201, and Swinburne University of Technology. We also would like to thank Fiona Coath for the development of the initial version of MEReq.

References

[1] G.Kotonya., and I.Sommerville, *Requirement Engineering Process and Techniques*West Sussex,England. John Wiley & Sons Ltd, 1998.

[2] H .Kassimand F.Ali, English communicative events and skills needed at the workplace: *Feedback from the industry*(2010), 168-182.

[3] Z.Shukur., A. M. Zin. A.Ban, andS.C.Ping, A Tool for Translating a Software Specification Written in Malay into a Formal Statement in Z, *Asian Journal of Information Technology* (2006),272-279.

[4] F.Calefato,F.Lanubile. and P.Minervini, Can Real-Time Machine Translation Overcome Language Barriers in Distributed Requirements Engineering, *5th IEEE International Conference on Global Software Engineering 2010, IEEE Computer Society*(2010),257-264.

[5] M.Kamalrudin., J.Hosking. andJ.Grundy, Improving Requirements Quality using Essential Use Case Interaction Patterns*, 2011 International Conference Software Engineering, Honolulu, Hawaii*(2011).

[6] F.Calefato, F.Lanubile, and R.Prikladnicki,*A* Controlled Experiment on the Effects of Machine Translation in Multilingual Requirements Meetings, *6th IEEE International Conference on Global Software Engineering*(2011),94–102.

[7] M.Kamalrudin.,J.Grundy andJ.Hosking,, Tool Support for Essential Use Cases to Better Capture Software Requirements, *25th IEEE/ACM International Conference on Automated Software Engineering, Antwerp, Belgium* (2010).

[8] M.Kamalrudin. and J.Grundy, Generating essential user interface prototypes to validate requirements,*26th IEEE/ACM International Conference of Automated Software Engineering (ASE 2011) ,Kansas,USA*(2011).

[9] M.Kamalrudin.,J.Grundy andJ.Hosking,Managing consistency between textual requirements, abstract interactions and Essential Use Cases, *34th Annual IEEE International Computer Software & Applications, Seoul, Korea* (2010).

[10] M.Kamalrudin., J.Grundy and J.Hosking,MaramaAI: Tool Support for Capturing and Managing Consistency of Multi-lingual Requirements*, 27th Automated Software Engineering 2012 (ASE 2012), ACM: Essen, Germany*(2011).

[11] M.Kamalrudin., J.Grundy and J.Hosking, Supporting Requirements Modelling in the Malay Language using Essential Use Cases,*IEEE Symposium on VisualLanguages and Human-Centric Computing, IEEE, 2012. Innsbruck,Austria* (2012).

[12] I.Groher, A.Reder and A.Egyed, Incremental Consistency Checking of Dynamic Constraints,*Fundamental Approaches to Software Engineering, Editors. D. Rosenblum and G. Taentzer*(2010).

[13] C.Nentwich.,E.Wolfgang and F.Finkelstein, Consistency management with repair actions, *25th International Conference on Software Engineering 2003, IEEE Computer Society: Portland, Oregon*(2003).

[14] Z.M.Don, Processing Natural Malay Texts:a Data-Driven Approach*, TRAMES: In A Journal of the Humanities & Social Science*(2010), 90.

[15] J.Holtmann., J.Meyer and M.V.Detten, Automatic Validation and Correction of Formalized, TextualRequirements, *2011 IEEE Fourth International Conference on. Software Testing, Verification and Validation Workshops (ICSTW)*(2011).

[16] J.Santos, Generating Requirements Analysis Models from Textual Requirements, *First International Workshop on.Managing Requirements Knowledge 2008(MARK '08)*(2008).
[17] M.Kamalrudin., J.Grundy and J.Hosking, MaramaAI: Tool Support for Capturing and Managing Consistency of Multi-lingual Requirements, *27th Automated Sofware Engineering 2012 (ASE 2012), ACM: Essen(* 2012).
[18] F.Meziane, Generating Natural Language specifications from UML class diagrams,*Requirements Engineering*(2008) ,1-18.
[19] A.Lund.(1998), USE Questionnaire Resource Page. Available, http://usesurvey.com/IntroductionToUse.html, 2011.
[20] A.Blackwell,C.Britton,A.Cox,T.Green,C.Gurr,G.Kadoda,M.Kutar,M.Loomes,C.Nehaniv,M.Petre,C.Roast,C.Roe,A.Wong and R.Youn, Cognitive Dimensions of Notations: Design Tools for Cognitive Technology*, Cognitive Technology: Instruments of Mind, M. Beynon, C. Nehaniv, and K. Dautenhahn, Editors*(2001), 325-341.
[21] M. Kutar, C. Britton., J. Wilson, Cognitive Dimensions An Experience Report, *Twelfth Annual Meeting of the Psychology of Programming Interest Group, Memoria, Cozenza Italy*(2000), 81-98.