MaramaAI: Tool Support for Capturing and Managing Consistency of Multi-lingual Requirements

Massila Kamalrudin
Faculty of Information &
Communication Technology
Universiti Teknikal Malaysia
Melaka, Hang Tuah Jaya, 76100
Durian Tunggal, Melaka, Malaysia,
massila@utem.edu.my

John Grundy
Centre for Computing and
Engineering Software Systems
Swinburne University of
Technology
PO Box 218, Hawthorn, Victoria
3122, Australia
jgrundy@swin.edu.au

John Hosking
College of Engineering & Computer
Science
The Australian National University
Canberra ACT 0200
Australia
john.hosking@anu.edu.au

ABSTRACT

Requirements captured by Requirements Engineers are commonly inconsistent with their client's intended requirements and are often error prone especially if the requirements are written in multiple languages. We demonstrate the use of our automated inconsistency-checking tool MaramaAI to capture and manage the consistency of multi-lingual requirements in both the English and Malay languages for requirements engineers and clients using a round-trip, rapid prototyping approach.

Categories and Subject Descriptors
D.2.1 [Software Engineering]: Requirements/ Specifications-tools

General Terms
Design; Documentation

Keywords
Multi-lingual Requirements; Consistency; Rapid-prototyping

I. INTRODUCTION

Requirements are commonly expressed and elaborated in English natural language. Often they are also expressed in other languages according to the mother tongue of the country where a given project is located. These natural language requirements are a very "human-centric" modelling approach, readily usable by clients, requirements engineers and developers alike. Multi- lingual requirements are also common in countries where more than one language is in common use. Malaysia is an example of country that often uses both English and Malay languages at the same time in communication [1]. This kind of "code-switching" communication also happens in capturing and eliciting requirements between requirements engineers and clients, which may lead to inconsistency and other quality problems.

These issues motivated us to explore tool support for capturing multi-lingual requirements in English and Malay and managing the consistency between them. In previous work, we developed a tool, MaramaAI, that provides round-trip prototyping support for checking consistency of requirements written in English based on Essential Use Case (EUC) diagrams [2],[3],[4]. EUCs are semi-formal models which we automatically extract from natural language requirements and validate against known EUC patterns. The EUC models are mapped to an abstract Essential User Interface (EUI) prototype model and from there they are mapped to concrete User Interface (UI) views in the form of HTML form-based UIs. This round-trip prototyping support provided by the tool permits the requirements engineer (RE) and client to walk-through the formalised requirements together and to capture, validate and confirm the consistency of these requirements.

Here we describe an extension of MaramaAI which supports capture and review of requirements in both English and Malay languages.

II. OUR APPROACH

Figure 1 illustrates our previous English language only approach [2,3,4,5]. In our new work, extraction of EUCs from text and consistency checking (steps 1-6) has been extended with essential interaction modelling "abstract interactions" in the Malay language. These are used to model requirements in the Essential Use Cases (EUCs) methodology. Our work allows the RE to automatically and traceably transform EUC models (Malay or English) to EUI prototypes using a novel EUI pattern library we previously developed [4] (step 7), but which we have extended with new EUI patterns in the Malay language. An example of two Malay essential interactions and their associated abstract interaction and EUI pattern is shown in Table 1.

These extensions to our earlier approach mean that traceability is provided throughout the process allowing any of the EUI components to be traced forward/back from/to

ASE'12, September 3–7, 2012, Essen, Germany.
Copyright 2012 ACM 978-1-4503-1204-2/12/09... $15.00..

the EUC model, abstract interaction or textual natural language requirement in either Malay or English. The EUI prototype can also be translated to a more concrete UI view, HTML web forms, by using a novel EUI pattern template library (step 8). Simple interaction with the generated HTML form is supported to illustrate how target system information input and output could work. This EUI model and concrete UI can be reviewed by the RE and client to validate and confirm consistency of the original textual requirements (step 9).

Table 1. Example of Essential Interactions and its associated Abstract Interaction and EUI Pattern

Essential Interaction	Abstract Interaction	EUI Pattern
Memasukkan id (insert id)	Mengesahkan diri (identify self)	ID
Memasukkan pin (insert pin)		

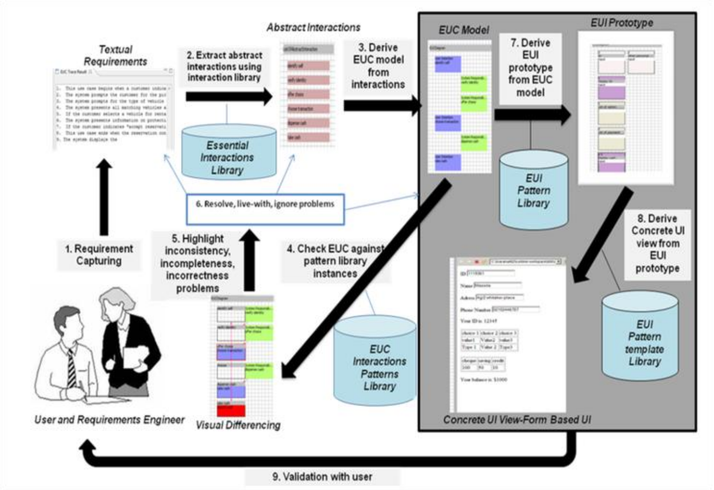

Abstract Interactions
Textual Requirements
2. Extract abstract interactions using interaction library
3. Derive EUC model from interactions
EUC Model
7. Derive EUI prototype from EUC model
EUI Prototype
Essential Interactions Library
EUI Pattern Library
6. Resolve, live-with, ignore problems
1. Requirement Capturing
5. Highlight inconsistency, incompleteness, incorrectness problems
4. Check EUC against pattern library instances
8. Derive Concrete UI view from EUI prototype
EUC Interactions Patterns Library
EUI Pattern template Library
User and Requirements Engineer
Visual Differencing
Concrete UI View-Form Based UI
9. Validation with user

Figure 1 End-to-end EUC and EUI prototyping approach (from [4])

III. TOOL SUPPORT

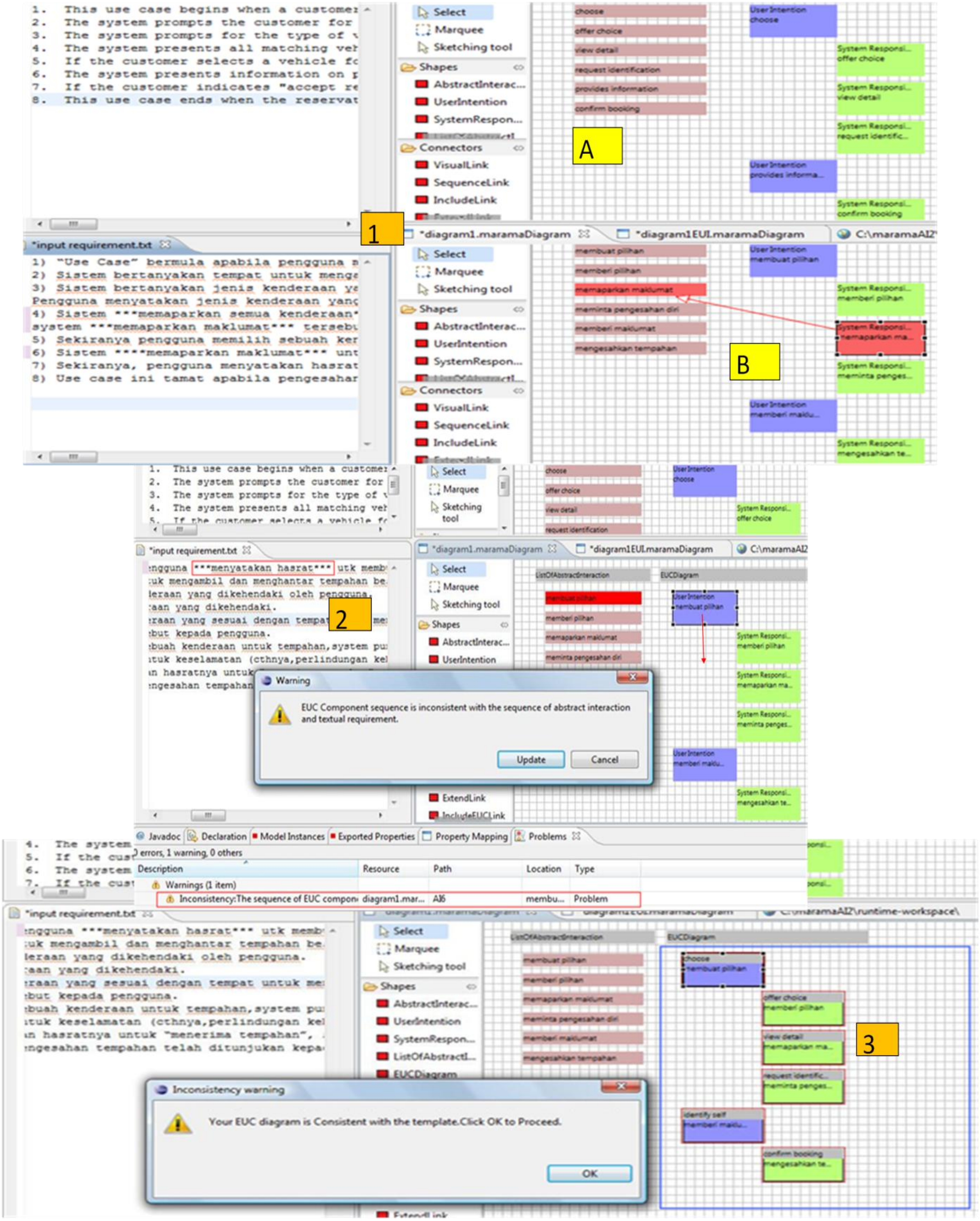
We have extended our original MaramaAI tool to support this approach to capture multi-lingual requirements in both English and Malay languages. The EUI prototype is modeled in a Marama diagram Editor, MaramaEUI and the concrete UI is presented in the form of an HTML page. Both are realized in the Eclipse IDE.

IV. USAGE EXAMPLE

We demonstrate the usage of our tool in capturing the requirements and managing the consistency of multi-lingual requirements written in both English and Malay languages with an example scenario from a set of requirements for reserving a vehicle from a rental company [5]. Massila, the requirements engineer, translates the requirements to the Malay language as the client is a Malay speaker. She then types in both requirements using the textual editor, editing each in a separate editor window, following the guidelines provided by the tool as shown in Figure 2(1). She then tries to capture both requirements as EUC models. She has the tool trace the abstract interactions and map the abstract interactions to EUC models for both English (A) and Malay (B) language requirements at the same time. She then has the tool check for the consistency of the Malay requirements component by having the tool to trace back the EUC component *"memaparkan maklumat"(view detail)* to its associated abstract interaction *"memaparkan maklumat" (view detail)* and essential interactions: *"memaparkan semua kenderaan" (show all vehicle)* and *"memaparkan maklumat"(view detail)* (shown in Figure 2 (B)).

While reviewing the captured requirements model, Massila disagrees with the sequence of abstract interaction *"membuat pilihan" (choose)* provided by the tool for the Malay requirements. Thus, she changes its location to the bottom of abstract interaction *"memberi pilihan" (offer choice)*. This action triggers the associated EUC component to automatically change its colour to red and the associated essential interactions "to be highlighted with "***". An inconsistency warning and problem marker also appear to inform her of the inconsistency and provides options to cancel or continue with changes (Figure 2 (2)). From these notifications of inconsistencies, she thinks she has to ignore her intention and stay with the generated EUC model.

Being a novice requirements engineer dealing with multi-lingual requirements, Massila thinks she needs to further validate her Malay requirements correctness and completeness compared to the English best-practice EUC template. She looks through the pattern catalogue and chooses the EUC pattern "Reserve item" that appears to be very similar to the requirements. She matches the pattern to her EUC model and sees that her Malay interactions are correct and complete compared to the English best-practice template (Figure 2 (3)). She then chooses to keep her EUC model for captured requirements. After Massila is satisfied with the requirements components, she meets Ahmad to validate further the requirements and to confirm the consistency of her captured requirements with the earlier requirements provided by Ahmad. In order to allow Ahmad to better understand the requirements components, she then has the tool map the EUC model to an abstract prototype: EUI prototype (A) and also has the tool translate the EUI prototype to a concrete UI view in a HTML form (B) (Figure 3(1)). They can view the prototypes for both requirements. Then, she and Ahmad walk through both the UI and the requirements components as shown in Figure 3(2) for better validation. From their discussion, Massila and Ahmad are satisfied as together they could confirm the consistency and validate the requirements. They are also happy that they could finalise both requirements quickly and without delay.



```
1.   This use case begins when a custome
2.   The system prompts the customer for
3.   The system prompts for the type of v
4.   The system presents all matching veh
5.   If the customer selects a vehicle fo
6.   The system presents information on p
7.   If the customer indicates "accept re
8.   This use case ends when the reservat
```

*input requirement.txt

```
1) "Use Case" bermula apabila pengguna m
2) Sistem bertanyakan tempat untuk menga
3) Sistem bertanyakan jenis kenderaan ya
Pengguna menyatakan jenis kenderaan yang
4) Sistem ***memaparkan semua kenderaan*
system ***memaparkan maklumat*** tersebu
5) Sekiranya pengguna memilih sebuah ker
6) Sistem ****memaparkan maklumat*** unt
7) Sekiranya, pengguna menyatakan hasrat
8) Use case ini tamat apabila pengesahan
```

```
ngguna ***menyatakan hasrat*** utk memb
uk mengambil dan menghantar tempahan be
leraan yang dikehendaki oleh pengguna.
aan yang dikehendaki.
raan yang sesuai dengan tempat untuk me
but kepada pengguna.
buah kenderaan untuk tempahan,system pu
tuk keselamatan (cthnya,perlindungan kel
n hasratnya untuk "menerima tempahan",
ngesahan tempahan telah ditunjukan kepa
```

Warning

EUC Component sequence is inconsistent with the sequence of abstract interaction and textual requirement.

Update Cancel

Problems

0 errors, 1 warning, 0 others

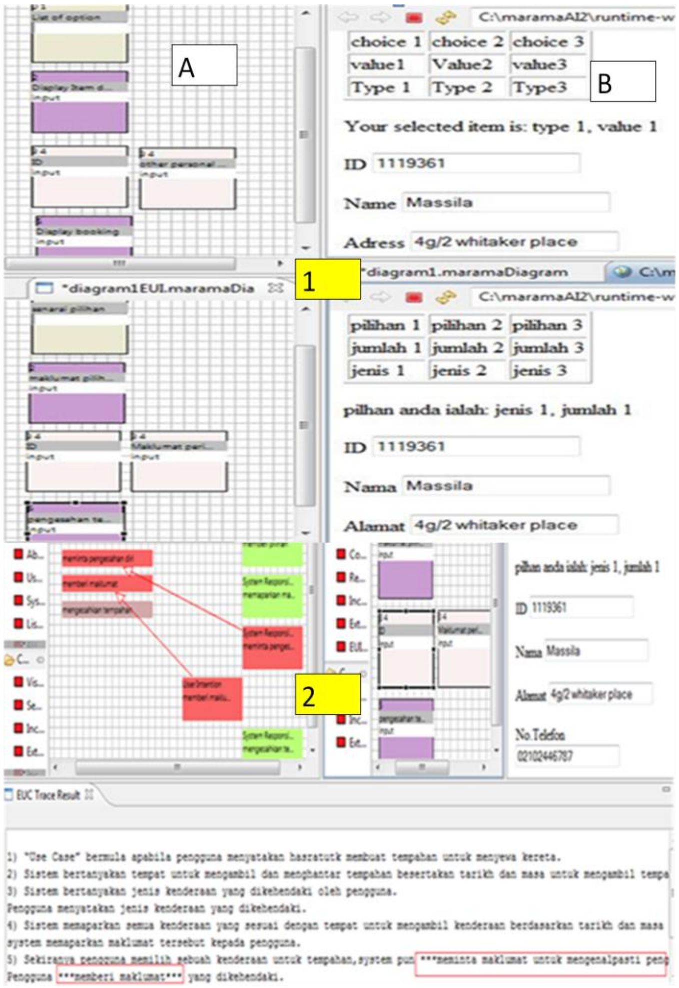
Description	Resource	Path	Location	Type
Warnings (1 item)				
Inconsistency:The sequence of EUC compon	diagram1.mar...	AI6	membu...	Problem

Inconsistency warning

Your EUC diagram is Consistent with the template.Click OK to Proceed.

OK

Figure 2 Capturing and Managing Consistency for Multi Lingual Requirements



choice 1	choice 2	choice 3
value1	Value2	value3
Type 1	Type 2	Type3

Your selected item is: type 1, value 1

ID 1119361

Name Massila

Adress 4g/2 whitaker place

pilihan 1	pilihan 2	pilihan 3
jumlah 1	jumlah 2	jumlah 3
jenis 1	jenis 2	jenis 3

pilhan anda ialah: jenis 1, jumlah 1

ID 1119361

Nama Massila

Alamat 4g/2 whitaker place

pilhan anda ialah: jenis 1, jumlah 1

ID 1119361

Nama Massila

Alamat 4g/2 whitaker place

No Telefon 0210446787

```
1) "Use Case" bermula apabila pengguna menyatakan hasratutk membuat tempahan untuk menyewa kereta.
2) Sistem bertanyakan tempat untuk mengambil dan menghantar tempahan besertakan tarikh dan masa untuk mengambil tempa
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
system memaparkan maklumat tersebut kepada pengguna.
5) Sekiranya pengguna memilih sebuah kenderaan untuk tempahan,system pun ***meminta maklumat untuk mengenalpasti peng
Pengguna ***memberi maklumat*** yang dikehendaki.
```

Figure 3 Mapping EUI prototype to Concrete UI (1) and Checking the Consistency of English and Malay natural language Requirements with the UI prototype (2)

V. CONCLUSION

We have described an extension to our MaramaAI tool to provide proof of concept, round-trip rapid prototyping support to capture and manage the consistency of multi-lingual requirements in both English and Malay languages. We use Essential Use Cases (EUC) to model the requirements and then map these Essential User Interface (EUI) prototypes to abstractly visualise the stakeholder requirements in either language. Our prototype can also translate EUI models to a more concrete UI view (currently in the form of an HTML page). We have conducted a preliminary user evaluation of our tool. This indicates that our automated tool support can improve multi-lingual communication in English and Malay languages between the requirements engineers and their stakeholder clients by using human-centric, auto-generated prototypes from semi-formal EUC models. These generated EUI and HTML forms help to provide a clearer picture of the requirements to the non-technical stakeholder client. It also allows confirmation of the consistency of the requirements captured by the engineers with a client's original natural language requirements by making the textual natural language requirements, abstract interaction, EUC model, EUI prototype model and HTML form mutually traceable to each other.

We are planning to conduct a formal evaluation of the effectiveness of our tool in improving the multi-lingual dialogue between the requirements engineer and the stakeholder clients. This evaluation will assess the effectiveness of our MaramaAI tool in assisting this communication and especially in aiding the confirmation of consistency of requirements captured by the requirements engineer using the tool with the client's original requirements.

ACKNOWLEDGMENT

We thank Jun Huh for his assistance with this research, which is funded by Ministry Of Higher Education Malaysia; Software Process and Product Improvement Project; & Universiti Teknikal Malaysia Melaka (UTeM).

REFERENCES

1. H. Kassim and F. Ali, "English communicative events and skills needed at the workplace: Feedback from the industry", English for Specific Purposes", vol. 29, pp. 168-182, 2010.
2. M.Kamalrudin, J.Grundy, J.Hosking, "Managing consistency between textual requirements, abstract interactions and Essential Use Cases" 34th Annual IEEE International Computer Software& Applications Seoul, Korea.2010
3. M.Kamalrudin, J.Hosking. J.Grundy, "Improving Requirements Quality using Essential Use Case Interaction Patterns," 2011 International Conference Software Engineering, Honolulu, Hawaii, USA, 2011.
4. M.Kamalrudin, J.Grundy, "Generating essential user interface prototypes to validate requirements". in 26th IEEE/ACM International Conference on Automated Software Engineering (ASE),Kansas,USA. 2011 .
5. M.Kamalrudin, J.Grundy., J.Hosking. "Tool Support for Essential Use Cases to Better Capture Software Requirements".25th IEEE/ACM International Conference on Automated Software Engineering.. Antwerp, Belgium. 2010.
6. G. Evans. Getting from use cases to code, Part 1: Use-Case Analysis., retrieved on January 2009: www.ibm.com/developerworks/rational/library/5383.html,