

See discussions, stats, and author profiles for this publication at: <https://www.researchgate.net/publication/311462055>

Mobility Requirements Engineering Tool (MoRE)

Conference Paper · September 2016

DOI: 10.1109/RE.2016.66

CITATION

1

READS

96

3 authors:



Xiaozhou Li

Tampere University

27 PUBLICATIONS 61 CITATIONS

SEE PROFILE



Biswa Upreti

Tampere University

1 PUBLICATION 1 CITATION

SEE PROFILE



Zheyang Zhang

Tampere University

45 PUBLICATIONS 270 CITATIONS

SEE PROFILE

Some of the authors of this publication are also working on these related projects:



Requirements analysis [View project](#)

Mobility Requirements Engineering Tool (MoRE)

Xiaozhou Li¹, Biswa Upreti², Zheyang Zhang³

School of Information Sciences
University of Tampere
Tampere, Finland

^{1,2}{li.xiaozhou.x, upreti.biswa.r}@student.uta.fi; ³zheyang.zhang@uta.fi

Abstract — The Mobility Requirements Engineering Tool (MoRE) is designed to facilitate the requirement analysis process of mobile app development towards the enhancement of mobile app mobility. The tool contains features of scenario creation and management, contexts and ways of interaction analysis and specification, as well as requirements change management.

Index Terms—Mobility Requirements, Scenarios, Situational Contexts, Interaction, Tools

I. INTRODUCTION

The mobility of mobile applications (apps) has not been adequately taken into account [1]. The changing environment of use, the status of the user at a given time, and the ways how the user interacts with the target mobile app, form the key factors that influence the perceived mobility an app [2]. Thus, by systematically analyzing these factors, we shall be able to specify and manage the requirements accordingly towards the enhancement of mobility. In addition, Scenarios are often used as a requirements elicitation technique to gather stories and to search for generalities [3]. Using scenarios as the approach to describing the situational contexts and the ways of interaction (WoI) [2] is also effective in acquiring and validating mobility requirements.

Aiming to facilitate the mobile app development process in acquiring and validating requirements towards mobility effectively, we hereby introduce the Mobility Requirements Engineering Tool (MoRE). The MoRE tool is a web application providing requirements acquisition and validation functionalities that enable the development team to specify contexts related requirements. It helps to enhance the mobility via situational contexts and ways of interaction analysis. The tool also enables scenario creation and management, which facilitates the contexts analysis and requirements acquisition. It provides an environment for mobile development teams to acquire, specify and validate requirements for the enhancement of an app's mobility.

II. SUPPORTING THEORY

The MoRE tool is developed on the basis of the user-app interaction reference model for mobility requirements analysis [2]. Furthermore, we adapt the previous mobility requirements analysis method by using scenarios to acquire context information and target features. The key concepts used in this tool are further explained as follows.

A. Situational Contexts

The situational contexts of a mobile app describe the spatial, temporal and social situations where the user interacts with

the app or a specific feature of it [2]. Shown in Fig 1, we define two temporal contexts, three spatial contexts, and two social contexts, when, therefore, 12 ($2 \times 3 \times 2$) unique situational contexts are obtained according to the rule of product. In addition, the primary situational contexts refer to the ones that must be taken into account for the app or a specific requirement.

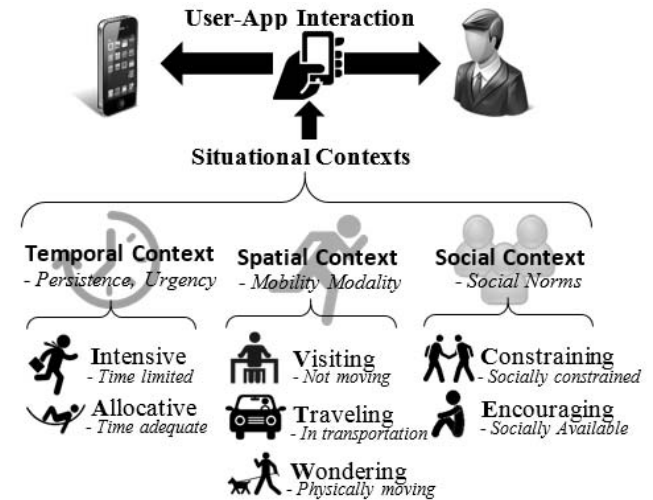


Fig. 1. Context Perspective Specification

B. Scenes

The scenes of the situational contexts refer to how the situational context is perceived in real life.



Fig. 2. Example Scene for IWE situational context

Shown in Fig 2, one of the scenes of situational context IWE is the user is running to work when he/she gets up late, as the user has limited time to spare, physically moving, and has no social norms constraining him/her from using mobile apps.

C. Ways of Interaction

The way of interaction describes how the user interacts with the app or a specific feature of it, which is determined by the persistence and obtrusiveness of the features of the app [2].

There are four ways of interaction, including, *accompanying*, *interrupting*, *intermittent* and *ignoring*.

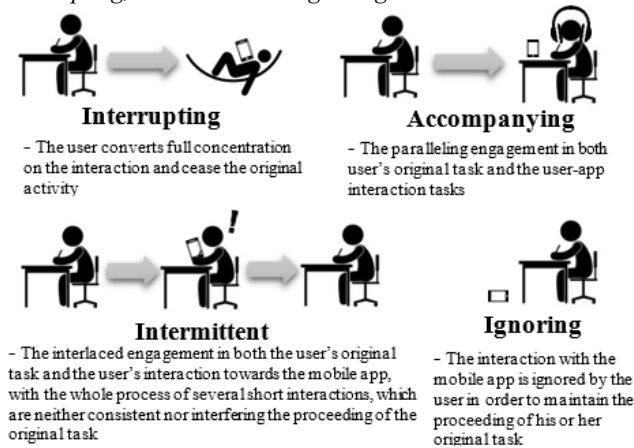


Fig. 3. Ways of Interaction

Furthermore, the expected ways of interaction (EWOI) indicates how the feature is expected to be interacted regardless of the contexts. The ideal ways of interaction (IWOI) refers to how the feature is ideally interacted in a specific situational context.

III. TECHNIQUES

The tool is developed in PHP with Laravel 5¹ when the database is maintained in MySQL. Other technologies, such as, HTML, CSS, and jQuery are also used. We have also used Bootstrap 3² for maintaining front-end designs. The IDE used for development is phpStorm³ with most of the development work done in Linux platform. We have also made it open source and the code is available at Github⁴.

IV. WORKING PROCESS AND FEATURES

The working process of the MoRE tool is described as follows (shown in Fig.4.).

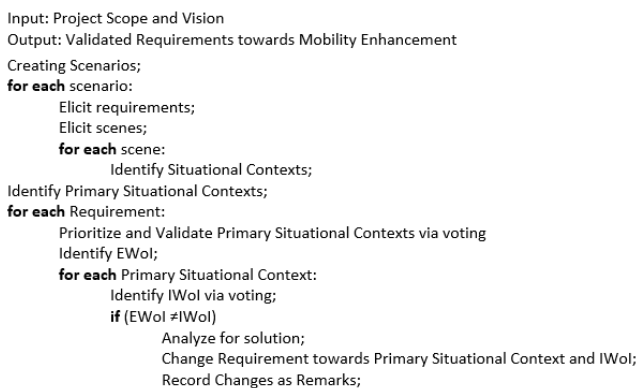


Fig. 4. MoRE Working Process

The tool creates a team workspace for collectively validate requirements via taking into account situational contexts and ways of interaction factors. Besides basic user profile and project management functionalities, the MoRE tool also include the following key features.

A. Scenario Creation and Management

A set of scenarios are created with the key information included to envision the use of the app. The tool enables development team to create and manage scenarios collectively.

B. Requirements Management

The project manager is able to create requirements based on the created scenarios, and keep track on the situational contexts and ways of interaction.

C. Situational Context Analysis

The tool enables team to categorize the primary situational contexts acquired from scenarios and prioritize them via the voting mechanism. Scenes can also be added to facilitate the analysis process by providing extended perspectives.

D. Ways of Interaction Specification

MoRE also enables the team to analyze the EWOI of each requirements and the IWOI of each primary situational contexts collectively. The team will detect the conflicts between IWOI and EWOI and adjust the according requirements towards the relevant primary situational contexts.

V. CONCLUSIONS AND FUTURE WORK

The MoRE tool provide a collective workspace for mobile development team to take into account the factors of situational contexts and ways of interaction. By doing so, the target mobile app product shall be designed to provide features that enable users to use it in a satisfactory way under those commonly encountered situations. However, the tool has not yet fully implemented with certain features lacking and further polishing. For example, the scenario creation feature shall be improved in order to obtain context and interaction information more effectively. Furthermore, the tool will be integrated with agile development tools, and shall be testified in mobile app maintenance as well.

REFERENCES

- [1] Kleinrock, L., 1996. Nomadicity: anytime, anywhere in a disconnected world. *Mobile networks and applications*, 1(4), pp.351-357.
- [2] Li, X. and Zhang, Z., 2015. A User-App Interaction Reference Model for Mobility Requirements Analysis. *ICSEA 2015*, pp. 170 – 177.
- [3] Sutcliffe, A., 2003, September. Scenario-based requirements engineering. In *Proceedings of the 11th IEEE international Requirements engineering conference, 2003.* (pp. 320-329). IEEE.

¹ <https://laravel.com/>

² <http://getbootstrap.com/>

³ <https://www.jetbrains.com/phpstorm/>

⁴ <https://github.com/biswaupreti/MobilityRequirements>