An Approach to the Use of Scenarios for Mobility Requirements Analysis

Xiaozhou Li[1], Zheying Zhang[2], Biswa Upreti[3], Timo Poranen[4], Yi Zhou[5]
School of Information Sciences
University of Tampere
Tampere, Finland
[1,3,5] {li.xiaozhou.x, upreti.biswa.r, zhou.yi.x}@student.uta.fi; [2,4] {zheying.zhang; timo.t.poranen}@uta.fi

***Abstract*— Mobile applications (apps) have been widely used and changing the life styles of people together with the permeation of various mobile devices. The way by which people use mobile apps in different situations has been seen as the core of them being mobile. However, it is very hard to predict and fully specify all different situations and to assure that the app provides comfortable and fluent interactions and services anytime and anywhere. In order to enhance the mobility of an app, we propose a scenario-based analysis approach for mobility requirements acquisition and validation. The approach takes into account a variety of situational contexts and their prioritization in analysis of ways of interaction implied by the app features under development. Furthermore, we develop a mobility requirements engineering tool (MoRE) to facilitate the process presented in the proposed approach. The prototyping tool helps stakeholders to analyze the mobility of existing app features. It is demonstrated and evaluated in requirements analysis for a voluntary firemen alarming application.**

***Index Terms* — Mobility, Requirements analysis, Scenario, Situational Context, Process, Tool.**

I. INTRODUCTION

Mobile apps have been changing people's life styles by enabling them to access the provided functionalities ubiquitously. The capability by which mobile apps providing pleasant and comfortable interactions "anytime, anywhere" for the users is referred to as the mobility [1]. The key factors related to mobility of a mobile app, besides the devices themselves, include the changing environment, the user and his/her status at that given time, as well as the way the user expects to interact in the app's operation [2]-[4][26]. However, the factors are seldom fully aware and taken into account in requirements engineering [5]. Requirements analysis on the mobility aspect seems mostly unguided or even non-existing in mobile app development, resulting in mobility and overall usability suffering [6].

To describe the imagined use of a mobile app in real world examples, scenarios can be often used for gathering stories and searching for generalities [7]-[9][11]. A scenario describes informally the imagined use of a future system, envisioned activities that new technology could enable, the possible event sequences of the systems, and the interaction events amongst the system, the user and their environment [8]-[11]. The key elements of scenarios, such as settings, goals, users, and actions, can largely provide insights in defining the interactions between the users and the target system, the context of using the system, as well as the different users and their goals [8][12]. It has been used as an analysis tool in strategic planning, user interface design, requirements engineering, software design, etc. [13][16]-[19][37]-[39].

A lot of research has contributed to scenario-based methods for requirements elicitation, specification and validation [13]-[23]. The methods provide outlined guidelines of using scenarios in requirements analysis and specification, such as [13][38][39]. However, many of them are lack of consideration of the importance of domain knowledge in creating scenarios for requirements analysis, and need to be tailored to the specified real-world usage. Research such as the taxonomy of scenarios in mobile application domain and the unbiased creation and analysis of scenarios seems rare. Some mobile app domain specific studies [20][21] focus on user behaviors and reactions but fall short on their relations towards various use contexts when other studies focusing on the context analysis oversee the rest [22][23].

In this paper, we tackle the issue on how scenarios help to acquire and validate mobile app requirements towards mobility. To answer the question, different factors influencing the receptive and pleasant use of an app shall be taken into account, including users, hardware systems and network, the physical environment, temporal context, tasks, social context, etc. [4][24]-[28]. Some of these factors such as the physical environment and the network and hardware settings have been studied in requirements analysis for self-adaptive software systems [29], while the situational context such as spatial, temporal or social context has not been thoroughly discussed in requirement analysis yet.

Hereby, we assume that users are willing to interact with the services offered by an app, and investigate the correlation between the situational contexts and the mobility requirements. We adapt the user-app interaction reference model [4], seeking for a concrete solution to analyzing both the contexts of mobile apps use and the user-app interactions towards the mobility enhancement. The proposed approach encompasses the process of mobility requirements analysis, the scenario analysis with a focus on situational context specification and prioritization, and the categorization of users' interactions in different situational contexts.

The remainder of the paper is organized as follows. Section 2 presents the motivation of using scenarios for requirements analysis in mobile apps domain. Section 3 reviews relevant studies on analyzing the contexts of mobile apps use and the ways of user-app interaction, on the basis of which we propose our approach to analyzing the context and the user-app interaction to enhance the mobility of a mobility app's feature. Section 4 describes the Mobility Requirements Engineering tool (MoRE) that is developed to facilitate the process presented in our proposed approach. Section 5 demonstrates how the tool is applied for requirements analysis in a voluntary firemen alarming mobile app, followed by discussion on relevant issues and future work in Section 6. Section 7 concludes the paper.

II. Scenarios in Requirements Analysis

A scenario is defined as an informal description of a situation in a system's environment and of a way that the system can be used [30], or a temporal sequence of interaction events between the target software and its environment (other systems or humans) in the restricted context of achieving some implicit purposes [31][32]. They are stories about people and their activities [33]. A typical scenario consists of four key elements: a setting, actors, goals or objectives, and actions, which have been emphasized in many other studies in similar forms (e.g. [8][12]). Scenarios are used to explicitly envision and describe significant user activities and to support analysis on situations of use before such situations occur [34]. Furthermore, scenarios can also evoke reflection about design issues, concretely fix interpretations and solutions, support stakeholder participation, enrich design knowledge and serve at multiple levels, from many perspectives and for many purposes [34]. For these reasons, they are often used for requirements acquisition and validation.

On the other hand, it takes a long process to specify the textual problem scenarios into usability specifications. Transforming from problem scenarios describing the unsatisfactory state of affairs to user interaction scenarios is then a key step [35]. By providing detailed activity-centered information as sketch of use, user interaction scenarios are used to vividly capture the essence of interactive design [35]. However, minimized attention has been paid to the activity of user interaction with more focus on the goals instead [35]. Many other scenario-based requirements analysis methods (e.g. [16][17]) have also limited consideration on the ways in which users interact with a specific system feature, as well as their relation towards the contexts, which is, however, critical to mobile application development.

A. Scenario-Based Requirements Engineering

To detect patterns in specific real world examples and extract the essence, scenarios are suitable for requirements engineering activities in gathering stories and searching for generalities [9][10]. Scenarios in terms of requirements engineering include facts describing an existing system and its environment including the behavior of agents and sufficient context information to allow discovery and validation of system requirements [8]. It emphasizes also the similar key elements of describing scenarios including settings, actors, goals and actions as the other studies [33][36]. In this study, concerning the domain of mobile app development, we adopt the terms context, users, goals, and interactions replacing these elements accordingly. Taking Fig 1 as an example, a scenario for a mobile phone user using instant messenger chatting with her friend is presented, distinguishing the key elements with different frames.

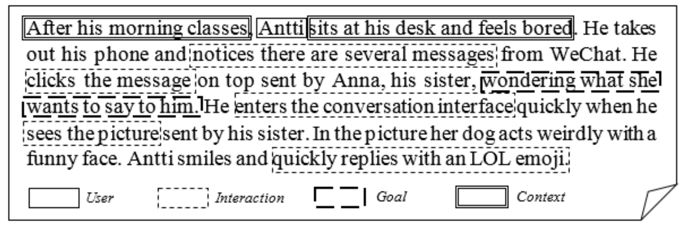

Fig. 1. Example of Scenario and Its Elements.

Users' action as a key element of scenarios has been studied by many focusing on its relation to the system rather than users' activity itself. Many other scenario-based requirements engineering methods also oversee the analysis on either that or the context element specification, and mostly both [17]-[19][39]. Nemoto et al. [40] gives an example of how to combine context specification and scenarios in requirements engineering. The study focuses on the physical environment as the context and does not cover the analysis of interaction between users and the target software.

Multiple methods and processes have been proposed to construct scenarios for requirements analysis in various forms. For example, SCRAM [8][13][37] presents an outline guidance for a scenario based requirements analysis. Using a combination of early prototypes, scenario scripts and design rationale, it guides users to validate design options for key points described in the scenario scripts and to document alternative designs in design rationale, which is also demonstrated in early prototypes. The scenarios are generated by elaborating pathways through a use case in two passes, i.e. normal behavior and abnormal one. In addition, Kaindle et al. [38] proposed a systematic process for selecting user interface elements on the basis of interaction tasks described in usage scenarios. The approach categorizes the tasks and maps them to widget classes for a GUI that shall support the execution of the scenarios. Obviously, the quality of the design is correlated with the completeness of given scenarios. The issue on how many scenarios are necessary to ensure sufficient requirements analysis is unaddressed in the research. In addition, several research papers [14][39] present a formal method for scenario analyses. Scenarios are translated into a set of executable state machines for simulation and validation. Specifically, the process presented by Hsia et al. [14] shows the scenario elicitation by constructing a scenario tree that describes and represents all scenarios for a particular user view. The scenario tree is further converted into a conceptual state machine.

B. Scenarios-Based Requirements for Mobile Development

Frequent using mobile apps in various contexts have largely changed the lifestyles of people and furthermore the ways of people perceiving the use of mobile apps [41]. On the other hand, the intrinsic mobility of mobile devices and the apps within have enabled users to use them ubiquitously [1][4][42]. Therefore, it includes a much larger set of situational contexts [4] where the apps are used, which additionally generates more possible scenarios.

In order to achieve the overall user experiences and quality, mobile apps are designed towards the enhancement of mobility, which is defined as the capability of providing receptive and pleasant services (interactions) acquired by users in spite of the changes in environments (contexts) [4][42]. The analysis on how to elicit and validate the requirements concerning different user behaviors in a variety of contexts is thus the key. The results, extracted and modeled from real world contexts, can be supported by applying scenarios analysis [9].

The contexts of mobile app use have also been studied widely. Different sets of context categorization have been presented in terms of mobile app usability and user experiences [25][43]-[45]. When considering contexts as the element of scenarios, we see the attitude and preference of users [46] and their goals and characteristics [47] as the intrinsic contexts when the time, space and sociality contexts [4][26][27] as extrinsic contexts or situational contexts. Based on the situational context model presented by Li & Zhang [4], the situational contexts can be further categorized into 12 groups, which can be instantiated in real world scenes in scenarios.

Meanwhile, limited studies have explored users' interaction with a specific feature of mobile apps concerning the characteristics of the feature and its influences towards the user at a given moment. Coping with that issue, the concept of ways of user-app interactions is carried out, indicating how user interactions are described based on the obtrusiveness and persistence of mobile app features [4].

Other studies also propose scenario-based requirements analysis methods for mobile app development, such as [22] and [23]. However, the scenario analysis of [22] provides only narrative scenario cases without identifying the analysis on the contexts and user behaviors within such scenarios. Giovanni et al.'s study [23], on the other hand, has taken into account the changes of environmental contexts. However, the differences between contexts and the connection towards the user experiences of mobile app features are overseen. In this study, we aim to adapt the user-app interaction reference model [4] with scenario facilitation in requirements analysis process for mobile apps. By doing so, we shall be able to concretely take into account both context and interaction, which facilitates the requirements acquisition and validation towards mobility enhancement.

III. Mobility Requirements Analysis Approach

Mobile apps are expected to be utilized in a wide range of situations, which requires them to be capable of providing satisfactory functionalities dealing with the possible contexts, which is defined as the mobility [4][42]. In order to enhance the overall mobility of mobile apps, the features shall be designed specially via analyzing the mobility related factors, the two important ones of which are the contexts and the ways of user-app interaction [4]. Using scenarios as artifacts to facilitate the process of eliciting and validating mobile app requirements is an effective way of achieving mobility, as scenarios have naturally taken into account the contexts and user interactions as the key elements [8][33]. Therefore we consider focusing on emphasizing the contexts and ways of interaction in the scenarios when analyzing requirements for mobile apps as the key in achieving mobility.

A. The Guided Scenarios towards Mobility

The textual narrative nature of scenarios have increased the difficulty in expressing the essential information explicitly. Thus in this study, we propose the guidelines of composing scenarios that not only express the key elements but also emphasize the perspectives of mobility related contexts and interactions.

The guidelines we propose for scenario creation include:

- Each scenario shall contain detailed description on the user's previous activity and his/her current physical and social status, which is called a scene.
- Each scenario shall describe in sequence of actions how the app is used.
- Different user groups shall be specified if any.

The aim of the guidelines is to guarantee that the scenarios created can convey the key information required to analyze the contexts in which users interact with an app and the ways of interaction, which will be elicited as requirements. Concerning the sample scenario shown in Fig. 1, the according key information summarized includes:

- User Group: *Antti (end user);*
- Scene: *After morning class, classroom, sitting, bored, alone (time allocative, stationary, socially alone);*
- Actions: *Click the notification on locked screen and open the app with the conversation shown on screen, send emoji to the other user via conversation.*

By using the scenarios, we can easily identify whether there are other situations where the user will act similarly with the app, and whether there are other ways of using it. Then by analyzing the information acquired from the scenarios, requirements analysts shall be able to derive a set of features and requirements, a set of scenes in which the app is used, a possible user group classification.

B. Situational Contexts Categorization

The context of the interaction between the user and the mobile app is defined as the information to characterize the situation between them, which contains multiple perspectives, including user profile, operating system, hardware system and network, physical context, temporal context, task context, and social context [24][25]. When considering the interaction between the user and the app feature as the essential, we thus pay minimized attention to the physical environment, device attributes and the user characteristics which adapting features in requirements level has limited influences to. We specify the contexts of our scenarios in three perspectives, including *temporal, spatial* and *social* [4].

The temporal perspective of context describes the sense of external time pressure of the user caused by the confliction or the accordance of user's goal and app's demands [26], with two values assigned: intensive and allocative [4]. The spatial perspective of context defines the current movement of the user further indicating the physical availability for the app usage, with three values of visiting, traveling and wondering [27]. The social perspective of context is interpreted as the social norms that constrain user from or encourage user into the interaction, with two values of constraining and encouraging [28]. The further explanation of each perspective value is shown in TABLE I.

TABLE I. CONTEXT PERSPECTIVE SPECIFICATION

Temporal	Intensive **(I)**	in urgent need of achieving other goals with limited spare time of interacting with the app (e.g. assignment deadline in 5 minutes)
	Allocative **(A)**	no urgent goal to achieve and is temporally available (e.g. idle at home)
Spatial	Visiting **(V)**	in a physically stationary status (e.g. sitting in class)
	Traveling **(T)**	in a transportation tool (e.g. sitting in a car)
	Wondering **(W)**	physically moving from place to place (e.g. walking in a building)
Social	Constraining **(C)**	the social norms require the user not to interact with the app (e.g. speaking to the president)
	Encouraging **(E)**	the social norms allow the user to interact with the app (e.g. drinking coffee alone)

By cross-combining the different perspective values of the situational contexts, we have then 12 unique situational contexts including *IVC, AVC, IVE, AVE, ITC, ATC, ITE, ATE, IWC, AWC, IWE, AWE* [4]. In order to ease the understanding of each situational context during the analysis process, scenes are connected with them, including the ones extracted from the scenarios. For example, the scene of "*sitting alone in classroom feeling bored*", extracted from the scenario in Fig. 1, falls into the situational context AVE, which indicates that the user is in a physically stationary status and temporally available, and is able to interact with the app.

Furthermore, in order to better understand the context of the target mobile app, the given situational contexts shall be prioritized due to how frequent the app is used in such a context. Based on the understanding of the scope and vision of the project, an initial list of primary situational contexts shall be identified out of the 12 contexts.

The prioritization of contexts differs for different features of the app. In order to explicitly prioritize the situational contexts, the requirements analyst team shall collectively decide the ranking for each requirements through voting. The voting can be organized with every stakeholder representative rating each situational context based on his/her understanding of its importance from one star to five. The obtained prioritization for each requirement shall be finalized via discussion referring to the primary situational context list.

C. Ways of Interaction Specification

As the users' action with mobile app is defined by how they interact with the app, we adopt the concept of ways of interaction describing the users' action [4][26]. And the mobility is thus reflected in the way in which users interact with the apps. According to the characteristics of the features, the ways of interaction between the user and the feature can be seen in four different types of interaction: *accompanying, interrupting, intermittent* and *ignoring*, with the further description as TABLE II.

TABLE II. SPECIFICATION ON WAYS OF INTERACTION

Ways of Interaction	Description
Accompanying	the paralleling engagement in both user's original task and the user-app interaction tasks
Interrupting	the user converts full concentration on the interaction and cease the original activity
Intermittent	the interlaced engagement in both the user's original task and the user's interaction towards the mobile app, with the whole process of several short interactions, which are neither consistent nor interfering the proceeding of the original task
Ignoring	the interaction with the mobile app is ignored by the user in order to maintain the proceeding of his or her original task

The ways of interaction can be determined by the persistence and obtrusiveness of the features of the app, and influenced by the situational contexts. The persistence of a mobile app feature indicates the sense of time required to achieve using this feature. The obtrusiveness of a feature is the tendency of using the feature interrupting the original activity of the user. Therefore, each of the mobile app features can be seen according to the categorization of obtrusive-persistent, unobtrusive-persistent, obtrusive-ephemeral, or unobtrusive-ephemeral.

TABLE III. EXPECTED WAYS OF INTERACTION

	Obtrusive	**Unobtrusive**
Persistent	Interrupting Accompanying	Accompanying Ignoring
Ephemeral	Intermittent Accompanying	Intermittent Ignoring

According to TABLE III, for each type of mobile app feature, the expected way of interaction (EWoI) can be decided. As both the persistence and obtrusiveness of app features are more or less subjective, the expected ways of interaction decision table shall be decided via team discussion on the characteristics of the app. On the other hand, for each situational context, ideal ways of interaction (IWoI) shall be also decided accordingly, indicating that the user shall feel comfortable using the app by such a way of interaction ideally. Thus, when there are conflicts between the EWoI and IWoI, especially when the context of IWoI is seen as the most important, the requirements shall be validated to address those.

D. Mobility Requirements Analysis Process

On the basis of the above discussion, we propose a mobility requirements analysis process to emphasize the importance of a variety of situational contexts when designing receptive and pleasant features for an app to achieve the overall user experiences and quality. The process of the mobility requirements analysis approach encompasses six unique steps, as follows.

Step 1. Scenarios Creation

A set of scenarios are created with the key information included to envision the use of the application.

Step 2. Feature Identification

A set of features shall be derived from the scenarios. The EWoI for each feature alone shall be extracted from the scenarios.

Step 3. Context Identification & Prioritization

For each feature identified in Step 2, the contexts of each related scenario shall be identified according to the situational context categorization. The identified contexts shall be also ranked based on the frequency of occurrence. The highly prioritized contexts and the scenes will be the most common situations in which a user interacts with the feature. They are firstly taken into account in requirements analysis.

Step 4. Interaction Identification & Specification

For each highly prioritized context, the IWoI shall be decided, indicating that the target feature shall be mostly used in this way in such a context. In addition, more scenes can be added to give concrete examples implying the IWoI.

Step 5. Conflicts Identification

Conflicts can be identified between the IWoI of any highly prioritized contexts and the EWoI of the feature. The conflicts shall be collected for further analysis.

Step 6. Requirements updated to enhance mobility

The feature with conflicting ways of interaction is further analyzed, and changes may be made to compromise the conflicts to ensure the receptive and pleasant use of the feature in the highly prioritized contexts.

The mobility requirement analysis approach encompasses the key activities of scenarios creation, requirements elicitation, situational context specification and prioritization, ways of interaction analysis, and requirements validation. Different from other requirements analysis processes, the approach focuses more on the analysis of situational contexts and the way of interaction between users and the app to discover and validate requirements. It emphasizes the changing environments, and shall enable the mobile development team to effectively analyze the requirements of the target app towards mobility enhancement. Importantly, the process shall be driven with quick response towards new scenarios and openness to changes. Team discussions shall be held constantly dealing with those changes while team manager shall make critical decision for the conflicts within.

IV. THE MOBILITY REQUIREMENTS ENGINEERING TOOL (MORE)

Following our discussion on mobility requirements analysis, we develop the MoRE tool to provide requirements acquisition and validation functionalities that enable the development teams to specify contexts related requirements enhancing the mobility. The tool enables the team to create projects and according scenarios, from which requirements are elicited. It also facilitates the process of situational context specification and prioritization, as well as ways of interaction specification, which enables the team to specify requirement and further provide app features towards primary situational contexts and comfortable ways of interaction. The tool also manages to help the development team in tracing requirements changes.

A. System Overview

The tool is developed in PHP. We have used Laravel 5[1], an open source PHP web framework in development. The database is maintained in MySQL. As the application is web based the rest of the technologies used are HTML, CSS, and jQuery. We have also used Bootstrap 3[2] for maintaining front-end designs. The IDE used for development is phpStorm[3] and most of the development work was done in Linux platform. As the project is a research work, we have made it open source and the code is available at Github[4].

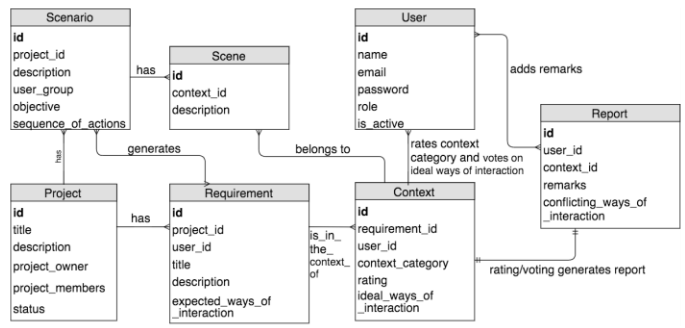

Fig. 2. The ER Diagram of the System.

The ER diagram shown in Fig 2 describes the system database architecture and the relations between the key entities. From the diagram we can see that scenarios helps the users in generating requirements and scenes. Scenes are instances a situational context. Furthermore, by rating and prioritizing the situational context and identifying, the conflicts between EWoI and IWoI are identified and reported for requirements validation.

B. Key Functions

The tool enables different user roles including administrator, project manager and developers. A user can login the system with registered email address and password. The administrator is able to control all the projects created in the system and to participate in projects as a developer or manager.

When a mobility requirement analysis project is created, the team members are allowed to add scenarios to the project based on its scope, as shown in Screenshot 1 of Fig 3. The scenarios will be discussed by the team in order to extract requirements. The project manager leads requirements analysis. New requirements are added, as shown in Screenshot 2 of Fig 3, on the basis of the team decision made during the discussion. In addition, the EWoI of the requirement will also be decided based on the characteristics of the target feature. The whole project team is able to browse the list of requirements, as shown in Screenshot 3 of Fig 3.

[1] https://laravel.com/

[2] http://getbootstrap.com/

[3] https://www.jetbrains.com/phpstorm/

[4] https://github.com/biswaupreti/MobilityRequirements

Furthermore, concerning each individual requirement, every project member is able to vote for the prioritization of the 12 situational contexts by rating with stars, and also the IWoI for each context (shown in Screenshot 4 of Fig 3). Meanwhile, the scene assigned to each context will be added by the owner of the according requirement. The scenes of the contexts will be taken into account by all team members in order to vote fairly.

The voting results are shown in the “Requirements Review” page (shown in Screenshot 5 of Fig 3). Conflicts between the IWoI of each context and the EWoI of the requirements will be identified. Furthermore, the project manager is enabled to put remarks concerning the list of specified requirements according to each conflict, based on the discussion of the whole team.

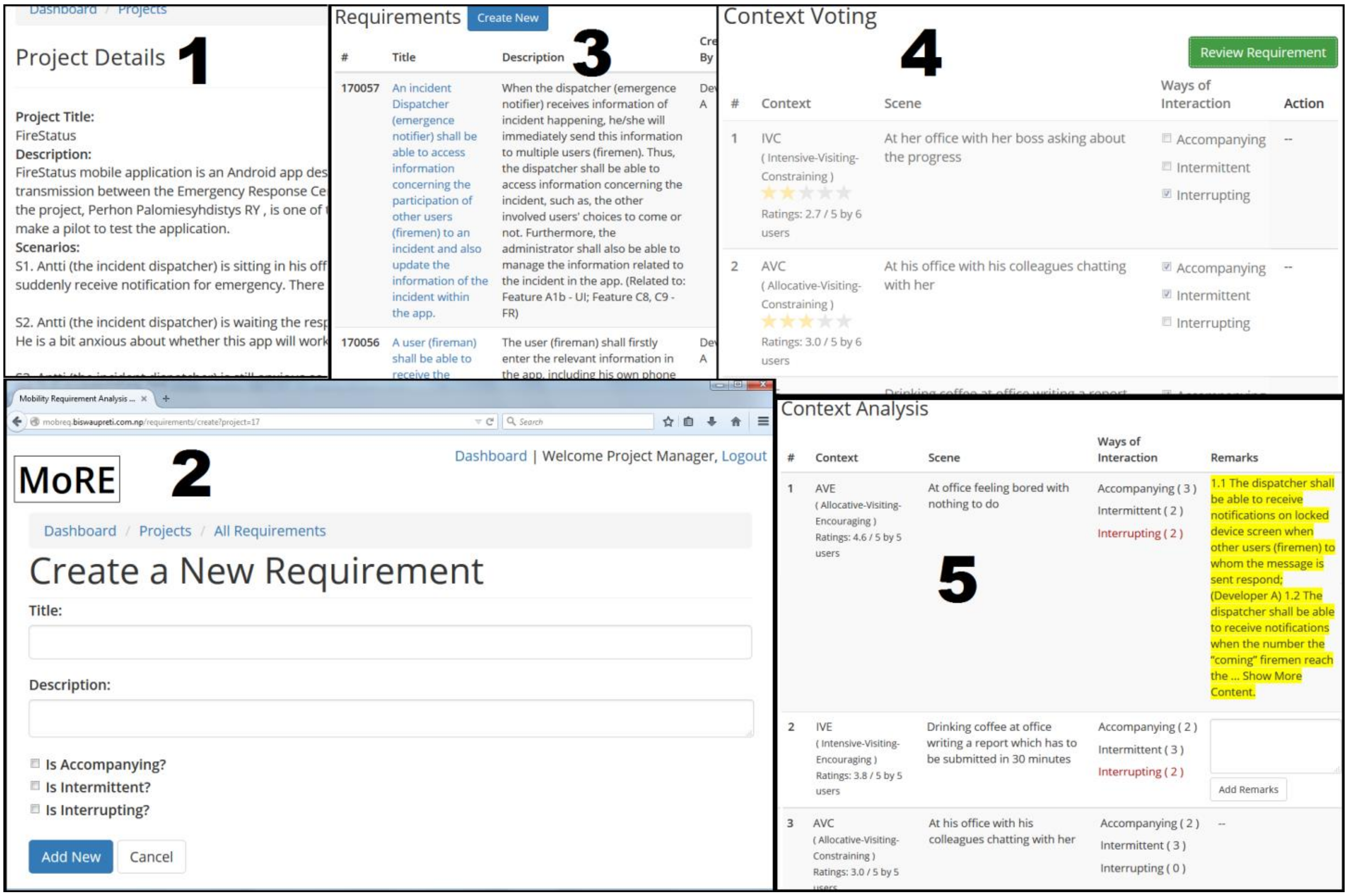

Fig. 3. The Screenshots of MoRE Tool with Key Features Specified.

V. Case Study

In this section we apply our proposed mobility requirements analysis approach into a real-life mobile app development case, facilitated by the MoRE prototypes that we develop. Our aim is to validate the usefulness and effectiveness of both the approach and the tool in requirements acquisition and validation activities towards mobility.

A. FireStatus Application

FireStatus is an Android app designed to facilitate the emergent message transmission between the Emergency Response Centre and Volunteer Fire Department. The client of the project, Perhon Palomiesyhdistys ry[5], is one of the Finnish Volunteer Fire Departments. Volunteer firefighters are well trained and ready to commit to serve the local communities for emergencies. In the volunteer fire department, the volunteer firemen are not present at the fire station, but at home, at work etc. When an alarm is announced, the volunteers come at different times, depending on where they are at that time. It is difficult for the dispatcher to know how many of them are coming and when they will arrive at the rescue site. The FireStatus app is an assistant in this matter. It enables the volunteer firemen to receive messages of emergency calls and respond quickly, and also facilitates the dispatcher to quickly acquire the information of responding firemen and how soon they can act on the emergencies.

The app was developed in the course Project Work[6] at University of Tampere in academic year 2014-2015. Its main features include receiving and responding to emergency messages

[5] http://www.perhonpalomiehet.fi/

[6] http://www.uta.fi/sis/tie/pw/previous-projects.html

(shown in Screenshot 1 of Fig 4), checking the responding situation of the incident call (shown in Screenshot 2 of Fig 4), and various settings such as setting the alarm sound, setting the keywords and incident dispatching number, and etc. (shown in Screenshot 3 of Fig 4).

The app is mainly used in local area, where Finnish is used as official languages. Screenshot 1 shows the Alarm (*HÄLYTYS*) tab, where it shows a testing fire alarm (*palohälytys*) message is received by a volunteer fireman, who can decide to come immediately (*HETI*), later (*MYÖHEMMIN*) or not coming (*EN TULE*) by clicking one of the three buttons. Screenshot 2 shows the participation information in the Participant tab (*OSALLISTUJAT*), which shows three firemen will come immediately (in the green bar) with their names and their skills shown in the gray area, another three comes in five minutes (in the yellow bar), one is unable to come (in the red bar) and 9 is without any response (in the white bar). Screenshot 3 shows the settings (*ASETUKSET*) tab, including the functions of changing alarm sound (*Hälytysäänet*), changing vibration (*Värinähälytys*) and so on.

Fig. 4. Screenshots of the FireStatus App.

B. Key Activities and Outcomes

The aim of this case study is to validate the usefulness of the mobility requirements analysis approach and our prototyping tool. Based on the documentation of the FireStatus project [51], we organize a simulating requirements analyst team and retrospectively analyze the requirements following the approach we propose by using the MoRE prototype. Our team consist of one project manager, who manage the process and control the outcomes of each individual phase, and four requirements analysts.

Before the requirements analysis starts, the project manager introduce the scope and vision of the FireStatus project to the rest of the team. The aim is to prepare the adequate background information concerning the user roles and their duties in emergent incidents, the process and activities of different user roles from the starting of the incidents till it is solved. The documentation of this project shall be used to ease the procedures of understanding. It will be also used during the process of the following mobility requirements analysis as references.

When all team members have familiarize themselves with the project scope, the team starts to create scenarios based on the acquired information. The scenarios are made following the guidelines proposed in the previous section, which demands the team to contain specific context related information, app features described by the actions of the user. A set of sample scenarios created are shown in TABLE IV (seen also in Screenshot 1 of Fig 3). The set of scenarios focuses on the incident dispatcher user group, who send emergence calls to firemen. And the scenarios describe a sequential changes in the user's interactions with the app and the situational contexts, which could commonly happen in real life.

TABLE IV. SAMPLE SCENARIOS

Scenario 1	Antti (the incident dispatcher) is sitting in his office browsing his Facebook page, when he suddenly receive notification for emergency. There is a fire alarm breaking out in a building 800 meters away. He immediately opens the app and sends the information to the list of volunteers with the keywords including emergency types, locations and number of people needed.
Scenario 2	Antti (the incident dispatcher) is waiting the responses from the volunteer firemen at his office. He is a bit anxious about whether this app will work as this is his first time to use it. He constantly checks the event screen when after only a few seconds, the first volunteer responds to come immediately.
Scenario 3	Antti (the incident dispatcher) is still anxious as one fireman is barely enough. He starts to walk around in his office. In another one minute, two other firemen reply to come immediately when also three others respond unable to come. He is then a bit relieved but still hopes at least two more firemen to come.
Scenario 4	Antti (the incident dispatcher) sits back and starts to browse websites randomly when he receives notification from the app indicating there are two more firemen will come in 5 minutes when one other responds unable to come. He believes that firemen will be enough to deal with the situation. He then opens the app and closes the event invitation.

When the scenarios are created, the team start to organize the scenarios with the key information, such as the target app features described in the scenarios and the scene where the according features are used. As a key outcome, a list of functional requirements shall be elicited and recorded in the tool. Meanwhile, based on the analysis on the persistence and obtrusiveness of the features, the EWoI of each feature is decided via team discussion and also recorded. A set of sample requirements created based on the scenarios are listed in TABLE V (seen also in Screenshot 3 of Fig 3).

TABLE V. SAMPLE REQUIREMENTS EXTRACTED WITH EWOI

ID	Requirements	EWoI
Requirement 170057	An incident dispatcher (message sender) shall be able to access information concerning the participation of other users (firemen) to an incident and also update the information of the incident within the app.	Intermittent Accompanying
Requirement 170056	A user (fireman) shall be able to receive the information of incidents via short message.	Intermittent Accompanying
Requirement	A user (fireman) shall be able to respond	Interrupting

170055	to the received incident notification.	

Then the team decides the list of primary contexts based on the understanding of the scope and the perceived scenarios. In this case, the dispatcher is considered as the one working mostly in an office, and use the app to send emergence calls to the others. Thus, the contexts related to the scenes of the dispatcher in office working environment shall be considered as the primary contexts

The outcome of the primary situational contexts are used as the references towards the following situational context prioritization for each individual feature. Meanwhile, additional scenes are also created together with the ones elicited from the created scenarios. Subsequently, each scene is identified according to the situational context categorization. The team guarantees that each situational context category is assigned with at least one context scene. The set of primary situational contexts accompanied with typical scenes are shown as TABLE VI (seen also in Screenshot 4 of Fig 3).

TABLE VI. PRIMARY CONTEXTS WITH SCENES

Context	**Scenes**
IVC	At her office with her boss asking about the progress
AVC	At his office with his colleagues chatting with her
IVE	Drinking coffee at office writing a report which has to be submitted in 30 minutes
AVE	At office feeling bored with nothing to do
ATC	Driving on an empty road to work in no hurry
AWC	Walking in the office building chatting with her coworkers.

Subsequently, the team analyzes each individual feature elicited from the scenarios by prioritizing the situational contexts accordingly. For each feature, each team member gives a rating on the importance of each situational context concerning this feature from one star to five. The context scenes are used as references facilitating the decision making. Furthermore, the IWoI for this feature used in the according situational contexts is also voted by each team member. When the result is calculated, the situational contexts are listed in the prioritized order while the conflicts between IWoI of the feature in each situational context and the EWoI of the feature are also identified.

As a result, the first six important situational contexts for Requirement 170057 as well as the conflicts in ways of interaction are shown in TABLE VII (seen also in Screenshot 5 in Fig 3).

TABLE VII. CONFLICTS OF INTERACTION FOR REQUIREMENTS 170057

Context	**Voting Results**	**Conflicts**
AVE	Rating: 4.6 /5 by 5 users	**Interrupting**
IVE	Rating: 3.8 /5 by 5 users	**Interrupting**
AVC	Rating: 3.0 /5 by 5 users	No conflicts
ATC	Rating: 2.8 /5 by 5 users	No conflicts
AWC	Rating: 2.8 /5 by 5 users	No conflicts
IVC	Rating: 2.5 /5 by 5 users	No conflicts

Shown in TABLE VII, in the first two situational contexts, which are seen as the most important ones, the conflicts of ways of interaction are identified. Shown in Screenshot 5 of Fig 3, there are two team members out of five consider the interrupting ways of interaction shall be taken into account when reviewing Requirements 170057 concerning the most important AVE and IVE context. The according solution towards the conflicts is to adding persistent or obtrusive functions to the feature. A series of remarks are made accordingly aiming to adapt the original requirements into one or a set of requirements that provides features the suitable ways of interaction. Based on the analysis on the remarks given by the team, Requirements 170057 is changed into TABLE VIII.

TABLE VIII. VALIDATED REQUIREMENTS 170057

Requirement 170057.1	The dispatcher shall be able to receive text, sound, and vibrate notifications on locked device screen when other users (firemen) to whom the message is sent respond.
Requirement 170057.2	The dispatcher shall be able to receive notifications when the number of the coming firemen reach the predefined limit, when the dispatcher shall close the incident.
Requirement 170057.3	The dispatcher shall be able to save relevant information to database concerning the process of the incident.

Requirement 170057.1 adds obtrusive notification as an extra feature, which evokes interrupting ways of interaction. Requirement 170057.2 contains the similar function. Requirement 170057.3, on the other hand, adds a persistent function, which will also evoke interrupting way of interaction. Therefore, by changing the original requirement, the conflict between EWoI and IWoI in the AVE context is solved.

VI. DISCUSSION

Via applying the mobility requirements analysis approach facilitated by MoRE, the mobile app development team is enabled to take into account both the situational contexts and the ways of user-app interaction, both of which are seen as the key factors enhancing the mobility of mobile apps. Specially, the approach contributes in the systematic analysis both the situational contexts and the ways of interaction, which provides an innovative perspective of mobile app requirements analysis. Furthermore, the approach and the MoRE tool largely ease the process of requirements acquisition and validation for mobile app development.

A. Tools for Requirements Analysis

There has been research on scenario based requirements analysis. The research results, however, remain process guidelines, and requirements tools rarely provide sufficient support in scenario and requirements analysis. Few research prototypes have been presented. For example, the CREWS-SAVRE [48] is a research prototype for systematic scenario generation and use. It supports the SCRAM method [37] for automatic generation of scenarios from a manually described use case. The scenarios are further used for semiautomatic validation of completeness and correctness of requirements. In addition, the scenario advisor tool [49] helps scenario generation by allowing users to write scenarios and by providing hints for scenario components to generate variations on existing ones; the requirements analysis support system (NDRASS) [50] supports consistency analysis between scenarios and requirements, and also support automatic synthesis of requirements models from a set of scenarios; etc. These research prototypes mainly support scenario generation and requirements validation, and have been analyzed and validated in experiments. Scenario generation is often a domain-driven process and needs domain knowledge [48][49]. If the tool is design as a general purpose tool for requirements analysis and validation in any application domain, it is hard to use in real-life requirements analysis. In addition, not every scenario takes the same role in requirements analysis. Overemphasis on one group of scenarios is likely to bias requirements validation results. The existing research prototypes seems lack of features to support scenario prioritization.

To cope with the issues mentioned above, the proposed mobility requirements analysis approach introduces the prioritization for categorized context, and uses scenarios as support. This approach enables the requirements analysts to obtain a better insight on the contexts of mobile apps and the ways of interaction concerning how the app is mostly used in these contexts. The understanding on both aspects further develop features that could be used comfortably in such contexts. And the MoRE tool facilitate the approach greatly via its handiness to access and use.

B. Limitation and Future Work

Despite of the concrete analysis on situational contexts, the complete description of the context of use is not fully presented by only using situational context. Other factors, such as, user characteristics, devices and network attributes, and physical environments, also influences the users' ways of interaction, which is not covered in this study. Furthermore, the tool prototype shall be further finalized by coping with the known quality functional deficits. Thus the future work of us shall contains efforts in tackling the issues mentioned above, and also in enriching the functionalities of the MoRE tool and integrating the approach and tool in a concrete development process, such as Scrum and Lean, for more optimized effectiveness.

VII. CONCLUSION

Mobility is one of the key attributes of contemporary mobile apps, and has yet been often taken into account during requirements analysis. The mobility requirements analysis approach we proposed provides concrete guidance in analyzing situational contexts, ways of user-app interaction, and the way of acquiring and validating requirements based on the analysis towards mobile app mobility. Furthermore, the MoRE tool we develop implement the approach and largely ease the process of mobility requirements analysis. Validated by the FireStatus app, our approach and tool shall large facilitate the future mobile app development projects achieving enhanced mobility.

REFERENCES

[1] Kleinrock, L., 1996. Nomadicity: anytime, anywhere in a disconnected world. *Mobile networks and applications*, 1(4), pp.351-357.

[2] Gorlenko, L. and Merrick, R., 2003. No wires attached: Usability challenges in the connected mobile world. *IBM Systems Journal, 42*(4), pp.639-651.

[3] Duh, H.B.L., Tan, G.C. and Chen, V.H.H., 2006, September. Usability evaluation for mobile device: a comparison of laboratory and field tests. In *Proceedings of the 8th conference on Human-computer interaction with mobile devices and services* (pp. 181-186). ACM.

[4] Li, X. and Zhang, Z., 2015. A User-App Interaction Reference Model for Mobility Requirements Analysis. *ICSEA 2015*, pp. 170 – 177.

[5] Cheng, B.H. and Atlee, J.M., 2007, May. Research directions in requirements engineering. In Proceedings of the 2007 *Future of Software Engineering* (pp. 285-303). IEEE Computer Society.

[6] Joorabchi, M.E., Mesbah, A. and Kruchten, P., 2013, October. Real challenges in mobile app development. In Proceedings of *International Symposium on Empirical Software Engineering and Measurement, 2013 ACM/IEEE* (pp. 15-24). IEEE.

[7] Nuseibeh, B. and Easterbrook, S., 2000, May. Requirements engineering: a roadmap. In *Proceedings of the Conference on the Future of Software Engineering* (pp. 35-46). ACM.

[8] Sutcliffe, A., 1998. Scenario-based requirements analysis. *Requirements engineering, 3*(1), pp.48-65.

[9] Sutcliffe, A., 2003, September. Scenario-based requirements engineering. In Proceedings of the *11th IEEE international Requirements engineering conference, 2003.* (pp. 320-329). IEEE.

[10] Potts, C., Takahashi, K. and Antón, A.I., 1994. Inquiry-based requirements analysis. *IEEE software, 11*(2), p.21.

[11] Jarke, M., Bui, X.T. and Carroll, J.M., 1998. Scenario management: An interdisciplinary approach. *Requirements Engineering*, 3(3-4), pp.155-173.

[12] do Prado Leite, J.C.S., Rossi, G., Balaguer, F., Maiorana, V., Kaplan, G., Hadad, G. and Oliveros, A., 1997. Enhancing a requirements baseline with scenarios. *Requirements Engineering, 2*(4), pp.184-198.

[13] Sutcliffe, A., 1997, January. A technique combination approach to requirements engineering. In *Proceedings of the Third IEEE International Symposium on Requirements Engineering, 1997.* (pp. 65-74). IEEE.

[14] Hsia, P., Samuel, J., Gao, J., Kung, D., Toyoshima, Y. and Chen, C., 1994. Formal approach to scenario analysis. *IEEE Software*, 11(2), p.33

[15] Dano, B., Briand, H. and Barbier, F., 1997, January. An approach based on the concept of use case to produce dynamic object-oriented specifications. In *Proceedings of the Third IEEE International Symposium on Requirements Engineering, 1997.* (pp. 54-64). IEEE

[16] do Prado Leite, J.C.S., Hadad, G.D., Doorn, J.H. and Kaplan, G.N., 2000. A scenario construction process. *Requirements Engineering*, 5(1), pp.38-61.

[17] Ralyté, J., Rolland, C. and Plihon, V., 1999, June. Method enhancement with scenario based techniques. In *Advanced information systems engineering* (pp. 103-118). Springer Berlin Heidelberg.

[18] Rolland, C., Souveyet, C. and Achour, C.B., 1998. Guiding goal modeling using scenarios. *Software Engineering, IEEE Transactions on, 24*(12), pp.1055-1071.

[19] Hickey, A.M. and Davis, A.M., 2004. A unified model of requirements elicitation. *Journal of Management Information Systems, 20*(4), pp.65-84.

[20] Yan, Z., Dong, Y., Niemi, V. and Yu, G., 2013. Exploring trust of mobile applications based on user behaviors: an empirical study. *Journal of Applied Social Psychology*, *43*(3), pp.638-659.

[21] Salo, M. and Frank, L., 2015. User behaviours after critical mobile application incidents: the relationship with situational context. *Information Systems Journal*. DOI: 10.1111/isj.12081

[22] Tran, K.N. and Nguyen, H.Q., 2013, December. SRE: A Scenario-based Requirement Exploration Process for End-user Mobile-Application Development. In *Proceedings of International Conference on Advances in Mobile Computing & Multimedia* (p. 559). ACM.

[23] Giovanni, P.D., Romano, M., Sebillo, M., Tortora, G., Vitiello, G., Ginige, T., De Silva, L., Goonethilaka, J., Wikramanayake, G. and Ginige, A., 2012, June. User centered scenario based approach for developing mobile interfaces for social life networks. In *First International Workshop on Usability and Accessibility Focused Requirements Engineering (UsARE), 2012* (pp. 18-24). IEEE.

[24] Abowd, G.D., Dey, A.K., Brown, P.J., Davies, N., Smith, M. and Steggles, P., 1999, September. Towards a better understanding of context and context-awareness. In *Handheld and ubiquitous computing* (pp. 304-307). Springer Berlin Heidelberg.

[25] Eshet, E. and Bouwman, H., 2014. Addressing the Context of Use in Mobile Computing: a Survey on the State of the Practice. *Interacting with Computers*, p.iwu002.

[26] Kakihara, M. and Sorensen, C., 2002, January. Mobility: An extended perspective. In *System Sciences, 2002. HICSS. Proceedings of the 35th Annual Hawaii International Conference on* (pp. 1756-1766). IEEE.

[27] Kristoffersen, S. and Ljungberg, F., 1999, August. Mobile use of IT. In *the Proceedings of IRIS22, Jyväskylä, Finland*.

[28] Chittaro, L., 2010. Distinctive aspects of mobile interaction and their implications for the design of multimodal interfaces. *Journal on Multimodal User Interfaces*, *3*(3), pp.157-165

[29] Oreizy, P., Gorlick, M.M., Taylor, R.N., Heimbigner, D., Johnson, G., Medvidovic, N., Quilici, A., Rosenblum, D.S. and Wolf, A.L., 1999. An architecture-based approach to self-adaptive software. *IEEE Intelligent systems*, (3), pp.54-62.

[30] Breitman, K.K., do Prado Leite, J.C.S. and Berry, D.M., 2005. Supporting scenario evolution. *Requirements Engineering*, *10*(2), pp.112-131.

[31] Liu, L. and Yu, E., 2001, May. From requirements to architectural design-using goals and scenarios. In *ICSE-2001 Workshop: From Software Requirements to Architectures (STRAW 2001) May* (pp. 22-30).

[32] Van Lamsweerde, A., 2000, June. Requirements engineering in the year 00: a research perspective. In *Proceedings of the 22nd international conference on Software engineering* (pp. 5-19). ACM.

[33] Carroll, J.M., 2000a. *Making use: scenario-based design of human-computer interactions*. MIT press.

[34] Carroll, J.M., 2000b. Five reasons for scenario-based design. *Interacting with computers*, *13*(1), pp.43-60.

[35] Rosson, M.B. and Carroll, J.M., 2009. Scenario based design. *Human-computer interaction. Boca Raton, FL*, pp.145-162.

[36] Go, K., Takamoto, Y. and Carroll, J.M., 2004, March. Designing a mobile phone of the future: requirements elicitation using photo essays and scenarios. In *Advanced Information Networking and Applications, 2004. AINA 2004. 18th International Conference on* (Vol. 2, pp. 475-480). IEEE.

[37] Sutcliffe, A.G. and Ryan, M., 1998, April. Experience with SCRAM, a scenario requirements analysis method. In *Proceedings of 1998 Third International Conference on Requirements Engineering, 1998.* (pp. 164-171). IEEE.

[38] Kaindl, H., Brinkkemper, S., Bubenko Jr, J.A., Farbey, B., Greenspan, S.J., Heitmeyer, C.L., do Prado Leite, J.C.S., Mead, N.R., Mylopoulos, J. and Siddiqi, J., 2002. Requirements engineering and technology transfer: obstacles, incentives and improvement agenda. *Requirements Engineering*, 7(3), pp.113-123.

[39] Araújo, S.G., Mesquita, A.C. and Pedroza, A.C., 2004. A scenario-based approach to protocol design using evolutionary techniques. In *Applications of Evolutionary Computing* (pp. 178-187). Springer Berlin Heidelberg.

[40] Nemoto, Y., Uei, K., Sato, K. and Shimomura, Y., 2015. A Context-based Requirements Analysis Method for PSS Design. *Procedia CIRP*, 30, pp.42-47.

[41] Skelley, T., Namoun, A. and Mehandjiev, N., 2013. The impact of a mobile information system on changing travel behaviour and improving travel experience. In *Mobile Web Information Systems* (pp. 233-247). Springer Berlin Heidelberg.

[42] Coursaris, C. and Hassanein, K., 2002. Understanding m-commerce: a consumer-centric model. *Quarterly journal of electronic commerce*, 3, pp.247-272.

[43] Schilit, B.N. and Theimer, M.M., 1994. Disseminating active map information to mobile hosts. *Network*, IEEE, 8(5), pp.22-32.

[44] Brown, P.J., 1995. The stick-e document: a framework for creating context-aware applications. *Electronic Publishing-Chichester-*, 8, pp.259-272.

[45] Korhonen, H., Arrasvuori, J. and Väänänen-Vainio-Mattila, K., 2010, December. Analysing user experience of personal mobile products through contextual factors. In *Proceedings of the 9th International Conference on Mobile and Ubiquitous Multimedia* (p. 11). ACM.

[46] Zhang, Z. and Zheng, X., 2007. User profiles and user requirements in mobile services. *Perspectives in Business Information Research-BIR'2007*, p.170.

[47] Thomas, K.W. and Velthouse, B.A., 1990. Cognitive elements of empowerment: An "interpretive" model of intrinsic task motivation. *Academy of management review*, 15(4), pp.666-681.

[48] Maiden, N.A., Minocha, S., Manning, K. and Ryan, M., 1998, April. CREWS-SAVRE: Systematic scenario generation and use. In *Proceedings of 1998 Third International Conference on Requirements Engineering, 1998.* (pp. 148-155). IEEE.

[49] Shin, J.E., Sutcliffe, A.G. and Gregoriades, A., 2005. Scenario advisor tool for requirements engineering. *Requirements Engineering*, 10(2), pp.132-145.

[50] Zhu, H. and Jin, L., 2000. Automating scenario-driven structured requirements engineering. In *The 24th Annual International Computer Software and Applications Conference, 2000. COMPSAC 2000.* (pp. 311-316). IEEE.

[51] Pekka Mäkiaho and Timo Poranen (eds.) Software projects 2014-2015, School of Information Sciences, University of Tampere (to appear)