Using Mobile RE Tools To Give End-Users a Voice

Norbert Seyff, Florian Graf, Neil Maiden
City University London
Centre for HCI Design
London EC1V 0HB, UK
n.seyff@soi.city.ac.uk florian.graf.1@city.ac.uk n.a.m.maiden@city.ac.uk

***Abstract*—Researchers highlight end-user involvement in system design as an important concept for developing useful and usable solutions. However, end-user involvement in software engineering is still an open-ended topic. Novel paradigms such as service-oriented computing strengthen the need for more active end-user involvement in order to provide systems that are tailored to individual end-user needs. Our work is based on the fact that the majority of end-users are familiar with mobile devices and use an increasing number of mobile applications. A mobile tool enabling end-user led requirements elicitation could be just one of many applications installed on end-users' mobile devices. In this paper, we present a framework of end-user involvement in requirements elicitation which motivates our research. The main contribution of our research is a tool-supported requirements elicitation approach allowing end-users to document needs in situ. Furthermore, we present first evaluation results to highlight the feasibility of on-site end-user led requirements elicitation.**

Keywords-end-user involvement; requirements elicitation; mobile computing

I. INTRODUCTION AND MOTIVATION

"The notion of 'user' cannot be precisely defined and therefore has no place in computer science or software engineering." [1]. Since this statement from Dijkstra, the view-point on end-user involvement in software engineering has changed. However, it is still an ambivalent topic. Researchers state that end-user involvement is a key concept in developing useful and usable systems [2], [3], [4], [5] but it still seems to have an inferior position in software engineering practice [2].

End-users can be involved in software engineering activities in manifold ways [6], [7], [8] and there exists no clear definition of what is meant by user involvement [8]. However, researchers discuss different levels of user involvement in software engineering [7], [8]. This includes end-users informing system design [7], [9], [10], [11], as well as end-users actually participating in the development of software systems [2], [12]. More recent developments focus on end-user programming [13], [14] [15] which tries to provide techniques and tools that allow end-users to create or modify software [16]. Such approaches include service mash-ups which aim to empower end-users in designing tailored service-based solutions themselves [17], [18]. However, some of those approaches have recently been discontinued (e.g. Microsoft Popfly [19]) and researchers highlight that alternatives are needed [20]. Although aimed at end-users these approaches are often built upon low-level programming concepts rather than on the concepts the end-users exploit for expressing their needs.

End-user involvement is particularly relevant in early software engineering activities such as requirements engineering (RE). The gathered end-user needs define what to build and therefore have a significant influence on the success of software projects [2], [21]. Several state-of-the-art requirements elicitation approaches involve end-users in requirements gathering (e.g. ART-SCENE [22], EasyWinWin [23]). Most typically, end-users are among the key stakeholders who are invited to participate in requirements elicitation workshops. These approaches often focus on discovering requirements that satisfy the needs of the majority of users [24]. However, novel software paradigms such as services-oriented computing suggest that identifying individual user needs [25] is essential to provide customized and tailored software systems [26].

Recent advances in mobile computing, such as the availability of sophisticated mobile devices (e.g. Google Nexus One, Apple iPhone) offer new possibilities for the involvement of end-users in requirements elicitation. Most end-users are familiar with mobile devices and people are using an increasing number of mobile applications. A mobile tool for requirements elicitation provides the potential to support end-users in documenting individual needs themselves anytime, anywhere.

In Section II, we discuss the research goal in more detail and present research questions. Section III reports on a framework identifying four different settings of end-user involvement in requirements elicitation. In Section IV, we report on the iRequire approach, which allows end-users to gather requirements themselves. Section V presents the iRequire tool, a mobile RE tool for end-users which uses the latest features of mobile devices to support end-users in documenting individual needs in situ. Furthermore, the tool automatically captures contextual information about their environment. In Section VI we discuss results from an initial evaluation study on the tool-supported approach, which highlight the utility of iRequire. In Section VII, we revisit the research questions and discuss threats to validity. Section VIII gives a conclusion and provides an overview on future research.

II. Research Goal and Questions

The goal of our research was to realize a tool-supported approach which enables end-users to document requirements themselves. The envisioned approach aims at strengthening active end-user participation in requirements elicitation by supporting end-users in documenting their individual requirements in situ. Based on this goal, more specific research questions (RQ) were identified to define the focus of our research:

RQ 1: Are end-users able to document their individual needs in situ with the help of mobile tools?

RQ 2: Can analysts transcribe end-user needs into well-defined requirements?

End-users can participate in requirements engineering activities in manifold ways. Our first research objective was to explore state-of-the-art strategies of end-user participation in RE and to establish a framework that highlights different settings of end-user involvement in requirements elicitation. This framework supported us in identifying particular settings where end-user participation in requirements elicitation could be strengthened.

Using the framework as a base, we aligned our research to meet the demands of end-users documenting requirements in the field. Requirements elicitation tools installed on mobile devices such as smartphones suggest the potential to support end-user led requirements elicitation. Therefore our second research objective was to develop a mobile tool supported requirements elicitation approach which allows end-users to capture their needs wherever and whenever they want. Furthermore, a third research objective focused on gathering contextual information to enrich requirements descriptions and to provide information on the end-users environment.

Our fourth research objective was to apply the tool-supported approach in order to explore end-user led requirements elicitation. We conducted an evaluation study in which end-users documented their individual needs in the field and used the results to investigate the utility of the tool-supported approach.

III. End-User Involvement in RE

In requirements elicitation, the end-user is often described as the primary source for identifying needs [2], [10] Researchers highlight the fact that end-user involvement improves the quality of requirements [2], [8], [12], [27] and that it varies in different requirements elicitation activities [22], [23], [28]. There exists a broad range of elicitation techniques such as brainstorming, interviews, workshops, and scenarios [29], [30], [31], [32] where end-users play an important role. However, most of these approaches focus on elicitation activities facilitated by requirements analysts who also document upcoming user requirements [22], [28], [33]. Researchers often describe end-user participation in requirements elicitation as informative [2], [7], this suggests a passive role for the end-user. Meanwhile, end-users are also more actively involved in requirements elicitation. For instance, agile approaches support customers in documenting needs themselves in the form of user stories [34], [35], [36].

Requirements elicitation activities are often performed out of the end-user's environment and placed into specialized settings such as workshops [22]. Researchers highlight the fact that this might have negative effects on gathering requirements because people are better at describing their needs when they have immediate access to social and material aspects of their daily life [3]. Blomberg et al. [3] discuss the importance of approaches which gather requirements in the users' environment [3], [33] rather than in workshops. This for example includes contextual inquiry [33] where analysts observe and interact with end-users to gather tacit and implicit knowledge. Such an approach provides potential benefits for requirements completeness and correctness [3], [33] and enables the analyst to better understand the existing work environment.

Analyzing state-of-the-art requirements elicitation approaches, we identified 4 different settings showing how end-users can be involved in requirements elicitation. These settings are basically defined by two dimensions: where requirements are gathered and who documents them (see Figure 1).

Setting I: represents approaches which conduct requirements elicitation out of the end-user's work context. These approaches rely on analysts to facilitate requirements gathering. Analysts aim to understand the problems and needs of end-users to then specify well-defined requirements. This setting is true for most workshop-based approaches. Examples include ART-SCENE [22] where analysts and stakeholders come together in workshops to walk-through structured scenarios in order to discover requirements.

Setting II: focuses on approaches where end-users themselves document upcoming requirements while being out of their work context. Examples include agile methods such as eXtreme Programming (XP) [34] and Scrum [37] which, for example, introduce the role of a customer who specifies requirements in the form of users stories. In contrast to well-defined requirements user stories are formulated in the everyday language of the end-user [36]. Researchers highlight the fact that the roles of end-users and customers are frequently overlapping [2] and future system end-users can act as customers in XP [38].

Setting III: represents approaches where analysts gather requirements in the work-place of future system users. This includes ethnographic approaches such as the coherence method [39], [40] and contextual inquiry [33], [41]. Following these approaches analysts observe end-users' work tasks to discover their needs on-site. Fieldwork enables analysts to understand end-users needs where they emerge. The analyst documents requirements using paper and pencil. Mobile RE tools such as the Mobile Scenario Presenter (MSP) provide more sophisticated support for on-site analysts and have been successfully used in several projects [28], [42].

Setting IV: The setting where end-users themselves document needs in their work context seems, so far, to be broadly neglected by requirements elicitation research and practice. A detailed literature review did not identify relevant work which would describe requirements elicitation approaches enabling end-users to document requirements in

situ. However, other disciplines such as Human Computer Interaction (HCI) have already introduced techniques such as cultural probes [43], where participants self-report about their activities (e.g. by using pen and paper to document their activities). Such techniques are typically applied to not influence end-users' tasks by on-site analysts and to gather information about tasks that are difficult to observe [43]. Dörner et al. [44] discuss the usage of cultural probes to enforce the collaboration between end-users and software developers.

Who Documents Requirements? / Where are Requirements Documented?	Analyst	End-user
Out of the End-users' Environment	Workshop-Based RE Approaches e.g. ART-SCENE	Agile Approaches e.g. XP, Scrum
In the End-users' Environment	Ethnographically Informed Approaches e.g. Contextual Inquiry, Mobile Scenario Presenter	?

Figure 1. Framework of End-user Involvement in Requirements Elicitation.

IV. A Tool-Supported Approach for End-User Led Requirements Elicitation

The results of our literature review and the presented framework highlight the need for research on in situ end-user led requirements elicitation. However, researchers argue that end-users are not able to document requirements themselves and rely on analysts to document their needs as well-defined requirements [45]. We agree with this view in general but follow the idea that requirements elicitation is an iterative process [46] and typically begins with the informal and incomplete description of needs. Therefore our approach intends to support end-users in capturing their individual needs. These initial requirements descriptions can provide the basis for consecutive refinement activities resulting in well-defined requirements descriptions.

A. Requirements for End-User Led Elicitation

The basic idea behind our approach is that mobile tools offer new possibilities to support end-users in documenting their individual needs in the field. However, little is known about how to support end-users in self-documenting their needs. In a first step we identified requirements that our end-user led elicitation approach has to fulfil. Therefore we generated a list of requirements which we considered important. We discussed and refined these requirements with 14 end-users. These discussions led to the following six key requirements:

Provide guidance and support: End-users are not familiar with requirements capturing. To avoid the need for lengthy briefings and to train end-users in documenting their needs our tool-supported approach should provide guidance and support for end-users.

Ubiquitous support: End-users need to go successfully about their everyday tasks while capturing their needs. They will not be able to focus on requirements documentation during their daily (work) activities. Therefore, our tool-supported approach has to be available anytime, anywhere and should be woven into the fabric of everyday life.

Fit unobtrusively: End-users should be able to document upcoming needs without drawing attention to themselves. For this reason, our approach is intended to enable them to blog their needs in an unobtrusive way.

Ad-hoc availability: While undertaking everyday activities, people do not have much time to blog needs. If end-users are not able to record a need immediately, they might forget about it. Therefore, a mobile tool for end-user requirements blogging should provide ad-hoc availability.

Support different ways of requirements documentation: End-users have different preferences in how they document their needs. Inspired by features of mobile devices our approach intends to enable end-users to capture their needs in different ways (e.g. by text-based descriptions and audio recordings).

Enriched requirements descriptions: Features of mobile devices enable end-users to capture information about their environment (e.g. by using mobile device's built-in camera). This information can be used to enrich and extend end-users' needs descriptions. We foresee that this additional information supports other people, such as analysts, in better understanding end user needs and the environment in which they emerged.

B. The iRequire Approach

We devised a tool-supported approach for end-user led requirements elicitation based on the reported needs. The iRequire approach requests that end-users document their requirements themselves and in situ (Setting IV). It is based on mobile tool support to facilitate end-user requirements blogging. iRequire enables end-users to capture upcoming needs in a structured manner. More specifically, the approach supports end-user led requirements blogging by defining three elicitation steps (see Figure 2):

Capturing contextual information: iRequire requests that end-users document information about their environment. This can be done using different media types, such as voice, video, pictures or natural language text. As the approach is based on mobile devices which often provide built-in context-sensing capabilities it demands going beyond end-user led documentation of contextual information.

Capturing end-user need: The main elicitation step requested by iRequire is the blogging of needs regarding an envisioned software system. End-users are required to document upcoming needs using text-based descriptions as well as different media types such as audio recordings. iRequire demands that descriptions, either natural language text or audio, are short and focused.

Capturing rationale and task: The iRequire approach also requests that end-users capture a rationale describing why a requirement is important to them. Furthermore, they are requested to document the task which they intend to support with the specified need. As for the documentation of needs, end-users can use natural language text descriptions and audio recordings to document a rationale and a task.

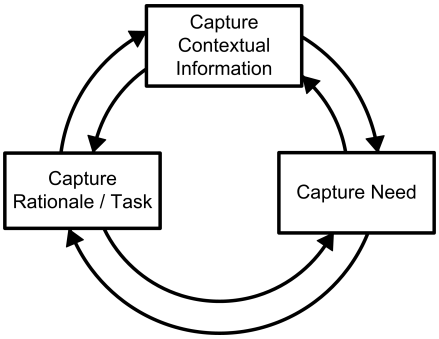

Figure 2. The iRequire Approach

The iRequire approach suggests to first capture contextual information, followed by the end-user's need and finally a rationale and task description (see Figure 2). We agreed on this sequence because it allows the description of needs referring to contextual elements (e.g. an end-user who takes a picture of a bus stop countdown display could document the need *I would like to have the same information on my mobile*). Furthermore, it seems natural to provide more information on a need (e.g. providing a rationale) after actually describing it.

However, a high degree of flexibility is essential for on-site elicitation approaches since end-users have to cope with a dynamic and changing environment. The iRequire approach foresees that end-users only document the kind of information they can provide at the time. Consequently iRequire requests that end-users are able to skip elicitation steps and go back to previous steps to add more information if required (see Figure 2). The flexible nature of the iRequire approach does not necessarily mean that end-users will not provide the information they could not document in the first place (e.g. due to time constraints). We foresee that documenting limited information cues (e.g. a picture of the environment) enables end-users to recall a need. These information cues provide a starting point for continuing the requirements documentation later and might prevent the end-user forgetting about a need.

Mobile RE tools implementing the iRequire approach can be made available for well-known mobile platforms (e.g. iPhone OS, Android, Windows Mobile). End-users could download and install these tools on their mobile phone. Therefore, iRequire would allow a high number of future system end-users to contribute to early requirements elicitation activities and would strengthen end-user participation in RE. While performing everyday task end-users apply iRequire to document their requirements in situ. Please note that the iRequire approach does not focus on end-user led brainstorming of needs. In contrast the aim of the iRequire approach is to enable end-users to capture needs which emerge in daily life. Therefore iRequire can be seen as an ubiquitous requirements elicitation approach where end-users still focus on their daily activities instead of the discovery of requirements. We foresee that after applying iRequire, the gathered end-user needs can, for example, be shared with analysts. They could analyze and transcribe them into well-defined requirements. Documented contextual information (e.g. pictures) can enrich requirements descriptions which can be used as input for further RE activities such as requirements negotiation.

V. iREQUIRE: A MOBILE RE TOOL FOR END-USERS

We decided to implement iRequire tool support based on modern smartphones such as the Google Nexus One, the iPhone and Windows Mobile Phones. By their nature smartphones fulfil key requirements of end-users for requirements blogging. Smartphones are small devices and can be used unobtrusively while end-users perform their daily tasks. They provide features such as built-in microphones and cameras enabling end-users to go beyond text-based requirements blogging. Smartphones do not have a start-up phase which ensures the ad-hoc availability of a mobile requirements blogging tool. Furthermore, they provide operating systems allowing end-users to install a broad number of different applications. An end-user requirements blogging tool could therefore be one of many applications available on modern smartphones and turn smartphones into mobile RE tools.

Taking advantage of the benefits of mobile devices we started to develop a prototype tool supporting the iRequire approach. Named after the approach, the iRequire application intends to be an easy-to-use mobile RE tool enabling end-users to capture their individual needs in situ. Following the iRequire approach, the application provides enhanced support and guidance for end-users and enables them to capture their needs in a structured manner.

A. Architecture and Design

The iRequire application prototype is currently available for Windows Mobile smartphones. We decided to develop for this platform as Microsoft Visual Studio provides a single development environment for programming in different .NET languages, such as C# [47]. Furthermore, we chose the Visual Studio Integrated Development Environment and Windows Mobile as we had already used it in previous research projects where it enabled fast tool prototyping.

The iRequire tool was built using C# and the .NET Compact Framework 3.5 [48], which is a subset of the .NET Framework tailored to run on mobile devices and enriched by libraries unique to mobile use (e.g. libraries to use the functionality of a mobile phone camera). The .NET Compact Framework provides an intermediate layer that allows access to built-in features of mobile devices (e.g. camera, GPS receiver) without considering manufacturer specific device drivers. To store identified needs and related data on the mobile device iRequire uses a Microsoft SQL Compact Edition Database [49].

The design of the iRequire makes use of novel interaction techniques provided by high-end smartphones. The iRequire user interface is optimized to be handled via a touch screen. We intended to provide a capable, but handy tool which is usable by smartphone users without tool introduction. Consequently we built the tool upon well-known interaction elements such as buttons and text-boxes.

End-users use natural language text to communicate needs [34]. Therefore, iRequire supports end-users to document needs in everyday language. Nevertheless, iRequire aims to support structured requirements documentation. It partially follows the VOLERE Template [50] and documents requirements identity, description, rationale and originator in the iRequire database.

The iRequire tool supports end-user requirements blogging via a wizard-like user interface which provides step-by-step guidance. More specifically, there is a four-step wizard for documenting a need. In the first step the end-user captures contextual information; in the second step the end-user documents a requirements description. The iRequire tool implements the order of elicitation steps suggested by the iRequire approach. However, following iRequire's idea of a flexible elicitation process, the tool enables end-users to skip any of the proposed elicitation steps. Furthermore, the wizard allows the end-user to go back to previous steps.

B. *Tool Features*

The following paragraphs describe the features of iRequire which support end-users in capturing their needs:

Capturing contextual information: The tool invites end-users to capture contextual information in the form of pictures of surroundings or objects that are related to their needs (see Figure 3). By pressing the *Take Picture* button end-users can use their mobile phone's built-in camera to capture information about their environment. With this feature, end-users can take several pictures they can refer to when documenting a need.

Documenting a need: This feature of iRequire enables end-users to blog their individual needs and ideas for an envisioned software system. More specifically, iRequire enables end-users to document text-based requirements descriptions (see Figure 3). When using this feature, end-users can document needs such as *I would like to know when the next bus is coming*. However, studies [42] have shown that using a mobile tool's (virtual) keyboard for entering text is not always comfortable and that audio recording is sometimes the preferred choice. Therefore, iRequire provides a dictaphone-like audio recording feature allowing the quick documentation of upcoming ideas and requirements. To prevent wordy and excessive descriptions of needs, iRequire limits textual descriptions and audio recordings to a maximum number of characters and a maximum time for recordings.

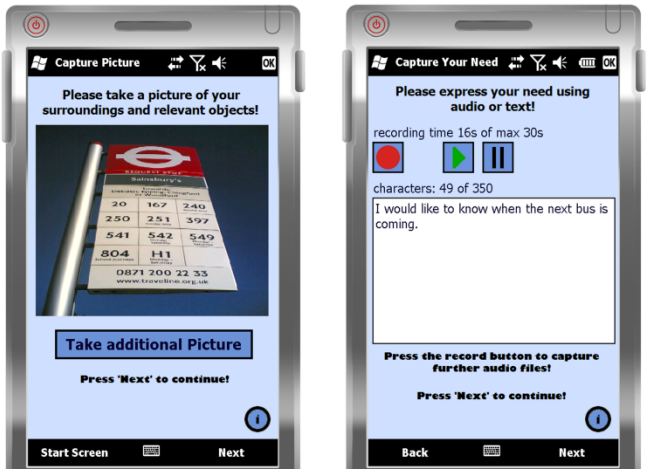

Figure 3. Figure 1 Taking a picture of the environment (left) and documenting a need (right) using iRequire

Describing the relevant task and providing a rationale: The iRequire tool enables end-users to describe needs in more detail. This includes documenting the task which the need is supposed to support. Furthermore, end-users can describe why this need is important to them by providing a rationale (see Figure 4). It is similar to the description of a need where end-users can capture this information via audio recordings or a short textual description. A possible task description for the above-described need could be *Such a feature would support my daily commuting* or *Waiting at a bus stop for a bus to my workplace*. A rationale for this need could be *I want to know when the next bus is coming in order to get my ticket ready* or *I could do something else if the next bus is coming late*.

Reviewing a summary: Before storing a captured need and contextual information, iRequire enables end-users to review the documented need and additional descriptions. It therefore displays a summary of the captured information to the end-user (see Figure 4) and requests an end-user's final commitment before the information is stored in the database. This step enables end-users to get an overview of the documented information and gives them the chance to add to or change the existing entries if required.

In addition to enabling users to document contextual information and needs, the iRequire tool is capable of automatically capturing relevant information about the end-user's environment. More specifically, the prototype is able to detect the end-user's position with the help of GPS and to store the GPS data together with captured needs. Furthermore, the iRequire stores a time stamp showing the time when a need was documented. These features enable the iRequire tool to describe more precisely the environment in which a need was captured without requiring user input.

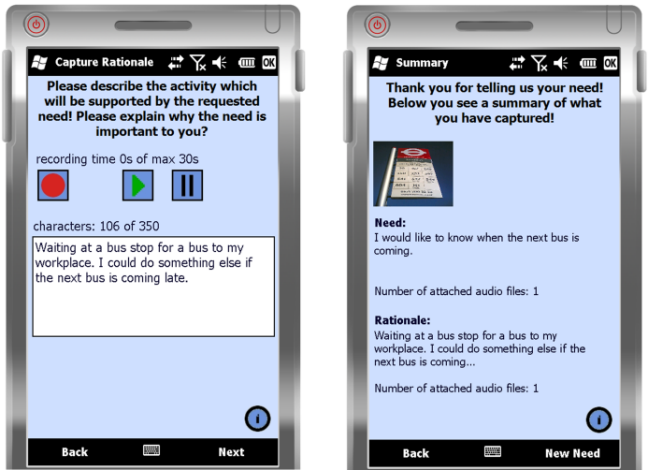

Figure 4. Figure 2 Capturing rationale / supported task (left) and summarizing captured information (right)

VI. EVALUATING IREQUIRE

To explore how the iRequire approach supports end-user led requirements elicitation we conducted an initial evaluation study. Our main goal was to investigate whether end-users can use iRequire to document initial requirements descriptions themselves while performing everyday tasks. Furthermore, we explored if requirements engineers are able to understand captured end-user needs and to transcribe them into well-defined requirements descriptions.

A. Method

In the first part of our study we invited 9 individuals of different ages (24 to 65 years), gender, profession and nationality to use iRequire and capture their needs in situ. None had any experience in requirements elicitation. The participants' knowledge of smartphones and mobile applications ranged between basic and advanced. Each evaluation was structured into 3 parts – briefing, evaluation and debriefing. During the briefing end-users were informed of the purpose of the study and the task they were intended to perform – using iRequire to discover individual needs for two mobile systems. One of these envisioned systems supports daily commuting, the other one shopping activities. However, it was open to participants to just document needs for one of the envisioned systems depending on their schedule. As iRequire was designed to be self-explaining the briefing meetings did not include a detailed tool introduction. During the evaluation each end-user applied iRequire in situ and documented upcoming needs while performing commuting and shopping tasks in the greater London area. More specifically, end-users were given the iRequire tool installed on a HTC Touch Pro 2 smartphone for an agreed period of time (1 to 3 days). During the debriefing the participants answered usability and utility questions on the iRequire tool and were asked about their experiences in structured interviews. After the debriefing meetings we analyzed the captured needs regarding quantitative and qualitative aspects. Furthermore, we sought to transcribe the captured end-user needs into well-defined requirements. We therefore read the needs documented in form of text and listened to the recorded audio files. We tried to find out if the pictures of the environment taken by end-users and the contextual information automatically captured by the iRequire tool provided support for understanding end-user needs.

B. Results

Qualitative feedback from the end-users was encouraging. All end-users claimed to be able to use iRequire to document upcoming needs in situ. Participants reported, based on a predefined scale, that iRequire supported them in documenting 80% of their needs immediately after they came into their mind. However, end-users mentioned that situations exist where the possibility to document needs is limited (e.g. on a crowded bus). All participants claimed that many of their needs were triggered by being in the field (e.g. missing a bus). They reported that using iRequire to record needs in situ did not affect their activities and that they could perform their commuting and shopping tasks as normal.

In total, the study participants documented 40 individual needs. 20 of these needs were related to the commuting support system and 18 to the shopping system. 2 were relevant for both systems as they combined support for commuting and shopping. Table 1 highlights the number of needs documented by participants and distinguishes between needs documented for the commuting system and shopping system. Example need descriptions included *Took a different bus would like to know if I am hasting [heading] the right way and where to get off and how long it will take* and *I need a warning function that looks at my shopping list and reminds me if I can get an item on my way if I pass by a shop selling it*.

The iRequire tool was made available to participants for an average period of time of 1.9 days. Participants were encouraged to keep the tool on their person to document needs in situ. During this time requirements capturing was not the participants' main task. An analysis of the iRequire log file revealed information on the usage of the tool. Participants captured needs within short time intervals with long times of inactivity in between. For example, one participant captured two needs within 3 minutes. She did not capture any for more than 24 hours but then captured the next two needs within 8 minutes. Within the short intervals of active documentation the stakeholders documented an average of 1 need every 4 minutes.

The analysis of the captured information (see Table 1) revealed that the majority of needs (70%) were documented using text-based requirements descriptions, while 30% of the needs descriptions were captured in form of audio files. The participants provided a rationale/task description for 70% percent of the documented needs. For instance, the rationale description for the shopping need example given above was *I forget sometimes to get groceries*. However, for this need the end-user did not provide a task description. Almost half (42%) of the rationale and task descriptions were captured in form of audio files. Furthermore, the participants enriched 67% of the captured needs by pictures of their environment or relevant objects. For example, an end-user attached a

picture showing the inside of a bus to a need requesting support for travelling on unknown bus routes. In addition to the information documented by the participants, the iRequire tool automatically enriched 55% of the documented end-user needs with GPS location information.

TABLE I. IREQUIRE EVALUATION STUDY RESULTS – RESUTLS SPLIT (COMMUTING / SHOPPING)

Participant	*End-user needs*	*Rationale/task description*	*Pictures*	*GPS context information*	*Transcribed requirements*	*Transcribed rationale*
1	1 / 4	0 / 2	1 / 0	0 / 0	3 / 9	2 / 7
2	0 / 5	0 / 5	0 / 5	0 / 0	0 / 15	0 / 7
3	2 / 2	2 / 2	0 / 0	2 / 2	2 / 2	2 / 2
4	3 / 2*	3 / 2	3 / 2	0 / 3	4 / 3	4 / 3
5	2 / 1	0 / 1	2 / 1	2 / 1	3 / 0	3 / 0
6	3 / 2	2 / 0	3 / 2	2 / 0	4 / 4	4 / 1
7	3 / 0	3 / 0	3 / 0	3 / 0	4 / 0	4 / 0
8	3 / 4*	0 / 4	0 / 2	2 / 4	6 / 4	4 / 4
9	3 / 0	2 / 0	3 / 0	3 / 0	4 / 0	2 / 0
	20 / 20	12 / 16	15 / 12	14 / 10	30 / 37	25 / 24

* contains requirements which were relevant for both systems but assigned to the shopping system

After end-users had captured their individual needs descriptions in the field, we focused on understanding these needs and transcribing them into well-defined requirements descriptions. End-users mostly blogged their needs in the form of short notes, documenting several ideas in one need description and sometimes mixed needs and rationales. Nevertheless, requirements analysts were able to understand the end-user needs and to transform them into well-defined requirements without requesting further information from the end-users. For example the need *Took a different bus would like to know if I am hasting [heading] the right way and where to get off and how long it will take* resulted into the following three well-defined requirements: (i) *The commuting system should be able to display the direction of the bus the user is currently travelling in*, (ii) *The commuting system should notify the user where to get off the bus* and (iii) *The commuting system should display the remaining travelling time to the users' destination*. The rationales for two of these requirements were included in the blogged need description. For example, the rationale for the first requirement was identified as *The end-user wants to know if she is travelling in the right direction*.

Requirements engineers needed 85 minutes to analyze the captured needs and transcribed the 40 end-user needs into 67 well-defined requirements. This means that an average end-user need contained more than 1.68 requirements. However, the 20 end-users needs on commuting were transcribed into 30 well-defined requirements (rate 1.5) and the 20 end-users needs regarding shopping triggered 37 requirements (rate 1.85). Three requirements were identified as duplicates. This means that in total we discovered 64 requirements relevant for the envisioned systems. We were able to transcribe all text-based user needs into well-defined requirements descriptions. However, 3 of the end-user needs documented via audio recordings could not be transcribed. Two recordings stopped unexpectedly after about 2 seconds. Another audio recording was useless due to the recorded background noise. Others were hard to understand but transcribable. As a result, in total 93% of text-based and audio-recorded user-needs could be transcribed into well-defined requirements descriptions. Furthermore, we were also able to identify a rationale for 71% of the well-defined requirements based on the information documented by the end-users.

During the transcription of blogged end-user needs we explored the pictures captured with iRequire. We rated 70% to be helpful to understand the situation in which an end-user need emerged (e.g. a picture of a bus stop which had been attached to a need regarding support for commuting by bus). The GPS location information automatically captured by iRequire provided relevant information about the documentation of needs in some cases. For example, when an end-user documented several needs on commuting the attached GPS information revealed that he documented these needs while commuting by train.

C. Conclusions

This study investigated if end-users are able to document their needs with the iRequire tool without being facilitated by a requirements analyst. Furthermore, we explored if requirements analysts are able to transcribe the blogged end-user needs into well-defined requirements descriptions without further end-user involvement.

In debriefing meetings, all participants confirmed that iRequire supported them in capturing their needs in situ. Participants captured 40 individual needs in total. On average end-users applied iRequire for 1.9 days, capturing 2.3 requirements per day. In terms of productivity this number seems to be below the average generation rate of workshop-based and facilitated approaches [28]. However, end-user led requirements elicitation is based on different conditions and does not follow the same strategy as workshops. For example, it allows a high number of end-users over a long period of time to participate in RE activates at low costs.

The evaluations study revealed that most needs were documented with the help of text descriptions (70%) rather than audio recordings (30%). We would have expected a higher number of audio recordings. Participants explained that they would generally have preferred to audio record needs in the field. However, it often drew the attention of others and made them feel uncomfortable (e.g. while sitting on the bus), so they started to document needs using text descriptions. As a second reason for using text-based need descriptions instead of audio recording end-users mentioned the high background noise in some environments. Participants further reported that in situations where they were in a confined public space they just documented limited information cues using text. In such situations they provided more detailed information on the need at a later point. This behaviour could explain the higher number of audio-recorded rationale and task descriptions compared to end-user needs.

The analysis of the gathered end-users needs revealed that, on average, needs on commuting support were transcribed into 1.5 well-defined requirements while end-user needs on shopping could be transcribed into 1.85 requirements. Participants highlighted the fact that, in general, documenting needs for the envisioned commuting support system was more challenging as commuting tasks are often performed in a noisy and crowded environment. This might be a reason for more focused need descriptions in the commuting domain.

Although we expected significant problems in transcribing end-user needs into well-defined requirements, it turned out that analysts were able to understand end-user needs and had few problems in transcribing them. Only the mixture of need, rationale and task description in some of the documented end-user needs challenged analysts and prevented the provision of rationale descriptions for some requirements. Contextual information gathered with iRequire supported analysts in understanding end-user needs and why they emerged. However, the study showed that automatically gathering contextual information only worked in 55% of the cases as GPS does not work indoors.

VII. Research Questions Revisited and Threats to Validity

As a result of the research we conducted and presented in this paper, we sought to answer the following two research questions.

A. Can end-users document their needs with a mobile tool?

To explore this question we designed and realized the tool-supported iRequire approach. It supports end-users in documenting individual needs while performing their daily tasks. Documenting a rationale and task description enriches end-user needs. Furthermore, the iRequire tool makes use of advanced smartphone features enabling end-users to take pictures of their surroundings. The tool goes beyond user driven documentation of contextual information by automatically capturing GPS data. As a proof of concept we presented an initial evolution study on iRequire. This study involved 9 end-users from different backgrounds. Most significantly, none had experience in or knowledge of requirements elicitation. Participants used iRequire to document individual needs in situ while performing daily commuting and shopping tasks. They applied iRequire and captured on average 2.3 needs per day. Furthermore, they were able to document additional information about the situation where the need emerged in the form of rationale/task descriptions and pictures of the environment. Our evaluation study revealed that end-users are able to document their individual needs in situ with the help of iRequire. Therefore we answer this research question with yes.

B. Can analysts transcribe end-user needs?

To explore the second research question we analyzed the captured end-user needs in order to transcribe them into well-defined requirements descriptions. Although end-users documented several ideas in one need description and sometimes mixed needs and rationales, analysts were able to transcribe all text-based need descriptions into well-defined requirements. The transcription of audio recordings was more challenging. Identified problems were mainly related to the high level of background noise on some recordings. One recording, in particular, could not be transcribed due to this issue. In total we were able to transcribe 93% of the gathered end-user needs. Transcribing needs into requirements was supported by additional information documented with iRequire. The pictures captured by end-users as well as the automatically captured GPS position helped analysts to better understand where a need emerged. Analysts claimed to have a high confidence in the correctness of the transcriptions. However, due to time and cost constraints we did not validate the transcribed requirements with the help of end-users. Therefore, the answer to research question two is only a tentative yes.

C. Threats to Validity

A threat to *construct validity* – are we measuring what we mean to measure – is the potential bias caused by the two domains (commuting, shopping) selected for requirements elicitation? This could mean that our study may under represent the construct. However, both domains were chosen carefully as tasks performed in these domains show typical characteristics of end-users' daily activities (e.g. a high degree of dynamics).

One threat to *internal validity* – are the results due solely to our manipulations – is that we did not conduct our study in a controlled environment? This means, that our results have to be interpreted with care. However, controlling end-user behaviour in the field was not attempted. Instead we sought results with ecological validity from a study in naturalistic settings. A further threat is that the transcription of end-user needs was not done by independent analysts. This could mean that transcribers were over motivated hence the possibility of biased results.

Regarding *conclusion validity,* we did not compute statistical significance when analyzing the results of our study. However, we sought to basically explore the feasibility and nature of in situ end-user needs documentation rather than providing detailed statistical data.

With respect to *external validity* – can we generalize the results – we investigated end-user led requirements elicitation with the help of iRequire. Different results might arise for different types of requirements acquisition tools and approaches. Our study revealed that analysts were able to transcribe end-user needs manually into well-defined requirements within a reasonable amount of time. However, this might change in projects involving a large number of end-users, which might result in overlapping, contradicting and repetitive need descriptions. Although end-users with different backgrounds and age were able to document needs with iRequire we see the small number of participants in our study to be a major threat to our results.

However, the uniqueness of iRequire and paucity of data about mobile RE tools for end-users means that these results provide valuable input for a setting broadly neglected by requirements elicitation research.

VIII. CONTRIBUTION AND FUTURE RESEARCH

In this paper we report on a solution supporting end-users in documenting individual needs themselves and in situ. We hope that our work will strengthen end-user participation in RE activities and inform the design of mobile RE tools for end-users. We claim 3 major contributions to RE knowledge:

- A framework identifying different settings of end-user involvement in requirements elicitation activities;
- An innovative tool-supported approach for end-user led requirements blogging;
- An initial evaluation study revealing that end-users are able to document individual needs without being facilitated by requirements analysts.

The lack of related work in RE means that our results, although preliminary, can offer new insights and depict new research opportunities. We particularly foresee that approaches such as iRequire could be applied to inform the development of software systems tailored and customized to suit individual needs of stakeholders (e.g. customized service-oriented systems [26]). Although developed for Windows Mobile, the iRequire prototype can be seen as a blueprint of a system that can be made available for a broad variety of smartphone devices. Therefore, the iRequire tool could guide researchers and practitioners in turning smartphones into mobile RE tools. However, third parties may decide to modify, enhance and improve the tool design.

Our future research will focus on the following issues:

Application Studies: Our initial tool evaluation revealed that end-users are able to blog their needs with the help of the iRequire prototype. Nevertheless, we need more sophisticated evaluation studies to explore the benefits and limitations of the approach and to provide more detailed usability and utility evaluation results. Furthermore, we will analyze the quantity and quality of the gathered needs and investigate the nature of individual end-user needs. This includes comparing iRequire results to the output of other requirements elicitation approaches (e.g. workshops). We also plan to apply iRequire in real-world projects to explore which domains and which kind of software projects can benefit from the iRequire approach.

iRequire Tool Extensions: We plan to adapt and enhance the iRequire tool prototype based on evaluation results and end-user feedback. In addition to user interface and feature improvements, we plan to make iRequire capable of gathering richer contextual information using technologies such as RFID and Bluetooth. For example, we intend to make the tool capable of sensing for Bluetooth signals of other mobile devices in order to detect the proximity of nearby people. This will allow iRequire to more precisely describe the environment in which a need was captured.

Distribution and Synchronization of End-User Needs: We consider individual end-user needs to be a starting point for further RE activities. Therefore, these needs have to be shared with other stakeholders. Currently, the end-user needs and contextual information are stored locally on the end-user's mobile device. End-users can use a desktop PC to access and distribute this information. Future research will explore how to distribute needs with other end-users and requirements analysts. We are planning to provide an advanced version of iRequire which will be able to distribute needs on-site by using mobile networks. This feature will allow end-users to immediately inform others about upcoming needs. This research also needs to address privacy issues that are raised by the automatic distribution of needs.

iRequire within the RE process: Informed by further studies our aim is to provide more guidance and support on how to use iRequire within RE activities. Furthermore, we will explore the manner in which iRequire can be integrated into existing RE approaches. A particular focus will be to investigate how end-user needs captured with iRequire can be used as input for further RE activities. We expect that involving a high number of end-users over longer periods of time will result in requests for automated transcription support. Therefore, future research will explore how to support requirements engineers in analyzing and transcribing end-users needs into well-defined requirements.

Requirements in Context: By using iRequire, our aim is to explore the influence of context on end-user requirements. We will concentrate on the analysis of captured contextual information to explore how the environment of end-users triggers their requirements and how context is related to needs. We will focus on ways to model and visualize captured contextual information in order to enrich requirements specifications. Furthermore we will explore if contextual information captured during in situ requirements elicitation can provide input for system design.

ACKNOWLEDGMENTS

This work was in part funded by the EU Marie Curie Fellowship Grant 236441 PIEF-GA-2009 and the City University Pump Prime Grant 90603CH.

REFERENCES

[1] E. W. Dijkstra, "Software Engineering: As It Should Be," Proc. 4th International Conference on Software Engineering, 1979, pp. 442-448.

[2] S. Kujala, M. Kauppinen, L. Lehtola, and T. Kojo, "The role of user involvement in requirements quality and project success," Proc. International Conference on Requirements Engineering, 2005, pp. 75-84.

[3] J. Blomberg, M. Burrell, and G. Guest, "An Ethnographic Approach to Design," in The Human-Computer Interaction Handbook: Fundamentals, Evolving Technologies and Emerging Applications, J.A. Jacko and A. Sears, Eds. Mahawah: Lawrence Erlbaum Associates, 2003, pp. 964-986.

[4] J. Karat, "Evolving the Scope of User-Centered Design," Communications of the ACM, vol. 40, 1997, pp. 33-38.

[5] A. Wilson, M. Bekker, P. Johnson, and H. Johnson, "Helping and Hindering User Involvement –A Tale of Everyday Design," Proc. Conference on Human Factors in Computing Systems, 1997, pp. 178-185.

[6] H. Barki and J. Hartwick, "Rethinking the concept of user involvement," MIS Quarterly, vol. 13(1), 1989, pp. 53-63.

[7] L. Damodaran, "User involvement in the systems design process-a practical guide for users," Behaviour & Information Technology, vol. 15(6), 1996, pp. 363-377.

[8] S. Kujala, "User involvement: A review of the benefits and challenges," Behaviour & Information Technology, vol. 22(1), 2003, pp. 1-16.

[9] K. Vredenburg, J. Mao, P. W. Smith, and T. Carey, "A survey of user-centered design practice," Proc. SIGCHI conference on Human factors in computing systems: Changing our world, changing ourselves, 2002, pp. 471-478.

[10] D. Schuler and A. Namioka, "Participatory Design: Principles and Practices," Hillsdale, NJ: Lawrence Erlbaum Associates, 1993.

[11] B. Ives and M. Olson, "User involvement and MIS success: A review of research," Manage. Sci. vol. 30(5), 1984, pp. 586-603.

[12] K. El Emam and N. H. Madhavji, "A Field Study of Requirements Engineering Practices in Information Systems Development," Proc. IEEE International Symposium on Requirements Engineering, 1995, pp. 68-80.

[13] A. Sutcliffe and N. Mehandjiev, "End-user development: tools that empower users to create their own software solutions," Communications of the ACM, vol. 47(9), 2004, pp. 31-32.

[14] H. Lieberman, F. Paternò, M. Klann, and V. Wulf, "End-user development: an emerging paradigm, " in End-User Development, H. Lieberman, F. Paternò, and V. Wulf, Eds. Springer Netherlands, 2006, pp. 1-8.

[15] C. Jones, "End-user Programming," IEEE Computer, vol. 28(9), 1995, pp. 68-70.

[16] G. Fischer, E. Giaccardi, Y. Ye, A. G. Sutcliffe, and N. Mehandjiev, "Meta-design: a manifesto for end-user development," Communications of the ACM, vol. 47(9), 2004, pp. 33-37.

[17] T. Loton, "Creating Google Mashups with the Google Mashup Editor," Lotontech Limited, 2008.

[18] D. Phuoc, A. Polleres, C. Morbidoni, M. Hauswirth, and G. Tummarello, "Rapid semantic web mashup development through semantic web pipes," Proc. World Wide Web Conference (WWW2009), Madrid, Spain, 2009.

[19] E. Griffin, "Foundations of Popfly: Rapid Mashup Development," Apress, 2008.

[20] D. Benslimane, S. Dustdar, and A. Sheth, "Service Mashups: The New Generation of Web Applications," IEEE Internet Compting, vol. 12(5), 2008, pp. 13-15.

[21] P. Tait and I. Vessey, "The effect of user involvement on system success: A contingency approach," MIS Quarterly, vol. 12(1), 1988, pp. 91-108.

[22] N. Maiden, "Systematic Scenario Walkthroughs with ART-SCENE," in Scenarios, Stories, Use Cases: Through the Systems Development Life-Cycle, I. Alexander and N. Maiden, Eds. John Wiley & Sons, 2004, pp. 161-178.

[23] P. Grünbacher and N. Seyff, "Requirements Negotiation," in Engineering and Managing Software Requirements, A. Aurum and C. Wohlin, Eds. Springer, 2005, pp. 143-162.

[24] S. Viller and I. Sommerville, "Social Analysis in the Requirements Engineering Process: From Ethnography to Method," Proc. IEEE Symposium on Requirements Engineering, Limerick, Ireland, 1999, pp. 6-13.

[25] A. Sutcliffe, S. Fickas, and M. M. Sohlberg, "Personal and Contextual Requirements Engineering," Proc. IEEE International Conference on Requirements Engineering, 2005, pp. 19-28.

[26] A. Granebring, P. Thilénius, and P. Révay, "SOA is a driver for Personalization," Proc. European Conference on Information Technology Evaluation, 2006, pp. 250-251.

[27] P. C. Chatzoglou and L. A. Macaulay, "Requirements Capture and Analysis: A Survey of Current Practice," Requirements Engineering, vol. 1(2), 1996, pp. 75-87.

[28] N. Seyff, N. Maiden, I. K. Karlsen, J. Lockerbie, P. Grünbacher, F. Graf, and C. Ncube, "Exploring how to use scenarios to discover requirements," Requirements Engineering, vol. 14(2), 2009, pp. 91-111.

[29] C. Coulin and D. Zowghi, "What Do Experts Think About Elicitation? – A State of Practice Survey," Proc. Australian Workshop on Requirements Engineering, Melbourne, Australia, 2005.

[30] E. W. Duggan, "Generating systems requirements with facilitated group techniques," Human-Computer Interaction, vol.18(4), 2003, pp. 373-394.

[31] J. A. Goguen and C. Linde, "Techniques for requirements elicitation," Proc. IEEE International Symposium on Requirements Engineering, San Diego, CA, USA, 2003, pp. 152-164.

[32] B. Nuseibeh and S. Easterbrook, "Requirements engineering: a roadmap," Proc. International Conference on Software Engineering, Limerick, Ireland, 2000, pp. 35-46.

[33] H. Beyer and K. Holtzblatt, "Contextual Design: Defining Consumer-Centered Systems," Morgan-Kauffman, 1998.

[34] K. Beck, "Embracing Change with Extreme Programming," IEEE Computer, vol. 32(10), 1999, pp. 70-77.

[35] K. Beck and M. Fowler, "Planning Extreme Programming," Addison-Wesley Longman Publishing Co, 2000.

[36] M. Cohn, "User Stories Applied: For Agile Software Development," Addison-Wesley Professional, 2004.

[37] K. Schwaber and M. Beedle, "Agile Software Development with Scrum," Prentice Hall, 2001.

[38] Z. Hussain, H. Milchrahm, S. Shahzad, W. Slany, M. Tscheligi, and P. Wolkerstorfer, "Integration of Extreme Programming and User-Centered Design: Lessons Learned," Proc. International Conference on Agile Processes and eXtreme Programming in Software Engineering, Sardinia, Italy, 2009, pp. 174-179.

[39] I. Sommerville, P. Sawyer, and S. Viller, "Viewpoints for Requirements Elicitation: A Practical Approach," Proc. Conference on Requirements Engineering, Colorado Springs, 1998, pp. 74-81.

[40] S. Viller and I. Sommerville, "Social Analysis in the Requirements Engineering Process: From Ethnography to Method," Proc. Symposium on Requirements Engineering, 1999, pp. 6-13.

[41] K. Holtzblatt, "Contextual Design," in The Human-Computer Interaction Handbook: Fundamentals, Evolving Technologies and Emerging Applications, J. A. Jacko and A. Sears, Eds. Mahwah: Lawrence Erlbaum Associates, 2003, pp. 942-963.

[42] N. Seyff, F. Graf, N. Maiden, and P. Grünbacher, "Scenarios in the Wild: Experiences with a Contextual Requirements Discovery Method," Proc. Requirements Engineering: Foundation for Software Quality, Amsterdam, 2009, pp. 147-161.

[43] B. Gaver, T. Dunne, and E. Pacenti, "Design: Cultural probes," IEEE Interactions, vol. 6(1), 2009, pp. 21-29.

[44] C. Dörner, J. Heß, and V. Pipek, "Fostering user-developer collaboration with infrastructure probes," Proc. International workshop on Cooperative and human aspects of software engineering, Leipzig, Germany, 2008, pp. 44-48.

[45] A. Rashid, D. Meder, J. Wiesenberger, and A. Behm, "Visual Requirement Specification In End-User Participation," Proc. International Workshop on Multimedia Requirements Engineering, Minneapolis, USA, 2006, p. 6.

[46] G. Kotonya and I. Sommerville, "Requirements Engineering: Processes and Techniques," Wiley, 1998.

[47] J. Plenderleith, "Microsoft Visual Studio 2008 Programming," Mcgraw-Hill Professional, 2009.

[48] P. Yao and D. Durant, "Programming .NET Compact Framework 3.5," Addison-Wesley Longman, 2009.

[49] SQL Server Compact 3.5, http://www.microsoft.com/Sqlserver/2005/en/us/compact.aspx.

[50] S. Robertson and J. Robertson, "Mastering the Requirements Process," Second Edition, Addison-Wesley Professional, 2006.