

Contextual Requirements Elicitation

An Overview

Thomas Keller (07-707-383)
t.keller@access.uzh.ch

Seminar in Requirements Engineering, Spring 2011
Department of Informatics, University of Zurich

Abstract. This paper gives an introductory overview on contextual requirements elicitation for software systems. Contextual requirements elicitation is defined as requirements elicitation that takes place at the customer's workplace. Ethnography and Contextual Inquiry are presented extensively as exemplary methods for contextual requirements elicitation. Their background, procedure, advantages and disadvantages as well as their integration into the requirements and software engineering process are discussed. This overview also raises critical issues concerning those methods and points out further research questions.

1 Introduction

The degree to which a certain software system meets the purpose for which it was intended can be taken as the main measure of success in software engineering [7]. Thus, requirements engineering as the discovery and further management of that purpose is indisputably an important part of software engineering.

One of the first tasks of a typical requirements engineering process is the elicitation of the requirements. Various methods are available for collecting requirements for software systems. Workshops, questionnaires, surveys, and the analysis of existing documentation are just some examples of methods currently available and in use [7].

The requirements can be collected at various places. It is conceivable for a requirements engineer to collect the requirements for a software system to be developed in his own office. He could read the existing documentation and draw on phone calls with the customer for further details. Other places where requirements can be collected are workshops and meetings which are attended by the customers and requirements engineers. However, requirements can also be elicited directly at the place where the system to be developed will be used. Distinguishing by the location where the elicitation of the requirements happens, the numerous methods for eliciting requirements can be split up into two groups. Requirements can be elicited apart from the customer's workplace or they can be collected directly at the workplace of the client.

Therefore, contextual requirements elicitation can be defined as requirements elicitation that takes place at the workplace of the customer. This means that

the requirements for the software system to be developed are collected in the context of the end user. They are captured where the system will later be used. The main goal of contextual requirements elicitation is usually to get detailed knowledge about the work area of the customer that would not be uncovered by the use of other, more conventional methods. Such contextual approaches have been practiced in requirements elicitation for at least twenty years by now and are still objects of research. One of the main reasons for the turn to contextual methods in requirements engineering was the growing support for the hypothesis that the development of many systems failed because their context had not been considered during development [5].

The purpose of the present paper is to give an introductory overview on contextual requirements elicitation for software systems. It is an introductory overview because not all of the various methods for contextual requirements elicitation are presented and not all recent research papers are reviewed. The covered range does not aim to be representative. However, it is sufficient to give a first impression and recognize some of the pending questions of the field. The paper presents ethnography and Contextual Inquiry as exemplary methods for contextual requirements elicitation.

The remainder of this paper is organized as follows. In section 2 ethnography is presented extensively as exemplary method for requirements elicitation for software systems. Contextual Inquiry is investigated as a second exemplary approach in section 3. Section 4 raises some critical issues concerning the reviewed methods and points out further research questions. Finally, the paper closes with a summary in section 5.

2 Ethnography

The term *ethnography* as used in requirements elicitation for software systems and in the present paper does not denote one clearly defined method. Rather, a bundle of different approaches fall under this concept. Those approaches have certain characteristics in common but differ in other respects. This section concentrates on that common subset of ethnographic methods in requirements elicitation and does not present a very specific ethnographic method.

Ethnographic methods have not been invented by requirements engineers. Their roots are in sociology and anthropology. Therefore, it might be useful to clarify what ethnography in those sciences is. This is done in section 2.1. Then, the application of the method for the elicitation of requirements is presented in section 2.2. This is followed by a discussion of the integration of ethnographic methods into the software engineering process in section 2.3. Their advantages and disadvantages are considered in the sections 2.4 and 2.5, respectively. Finally, the investigation on ethnography closes with a summary in section 2.6.

2.1 Preliminaries

As already mentioned, the origins of ethnographic methods lie in sociology and anthropology. The use of ethnography in those sciences can be called *classical*

ethnography. In contrast, ethnographic methods that are used to capture requirements or – more generally – are applied in the context of software engineering can be called *applied ethnography*.

The definition of classical ethnography is fuzzy and varied [1, 4]. The term is used to denote various approaches that involve fieldwork in one way or the other [4]. It can be said that ethnography is an observational method and tries “to present a portrait of life as seen and understood by those who live and work within the domain concerned” [6, p. 28]. Thus ethnography can be regarded as method that attempts to understand a certain situation as it is perceived by the actors of that situation.

The fact that there is no clear definition of what ethnography in its classical sense consists of is unsatisfying. Ball et al. have reviewed the relevant literature including articles and monographs to find out what classical ethnography is. They found the following ten consistently occurring features (adapted from [1], put in alphabetical order):

Historicism The observer aims to connect observations to a backdrop of historical and cultural contingencies.

Independence The observer aims not to be constrained by any pre-determined goal-set, mind-set or theory.

Intensity Observations are intensive and long-term so as to enable the observer to become immersed in the ongoing culture of the observee’s environment.

Openness The observer remains open to the discovery of novel or unexpected issues that may come to light as a study progresses.

Participant Autonomy The observees are not required to comply in any rigid, pre-determined study arrangements.

Personalization The observer makes a note of their own feelings in relation to situations encountered during data collection and analysis.

Reflexivity The observer adopts a reflective and empathetic stance in striving toward an understanding of the observee’s point of view; the observer taking account of, rather than striving to eliminate, their own effects upon the behavior of the observees.

Richness The observer studies behavior in all of its various manifestations such that data are gathered from a wide range of sources including interview, team discussions, incidental conversations, documents, and non-verbal interactions.

Self-reflection The observer acknowledges that any interpretative act is influenced by the tradition to which they themselves belong.

Situatedness Data are collected by a participant observer who is located within the everyday context of interest.

In classical ethnographic studies the degrees of subjectivity, reflexivity and self-reflection are high. Equally high are the observational openness and independence. The intensity in such studies can be called extreme, because some anthropological studies using ethnographic methods last several years [1].

Within the scope of this paper we are not especially interested in classical but in applied ethnography. The above list might be an interesting starting point

to find out what applied ethnography is. Some of the characteristics mentioned are relevant for requirements elicitation for software systems, others are not.

Historicism as the connection of current observations with historical and cultural facts can probably be neglected. Independence as the impartiality and purposelessness of the investigation is more than problematic for an ethnographic study in requirements elicitation that has the very specific goal to collect requirements. Intensity is another point which might be quite different in a sociological study than in a requirements elicitation under time pressure.

It is not surprising therefore that most design studies violate a significant part of these characteristics [1].

Ball et al. [1] offer three reasons why applied ethnography differs from classical ethnography.

First, the intensity of classical ethnography is often not cost-effective for most projects. Classical ethnography aims to know as much as possible about the object of research and therefore depends on long-term and continuous observations. This is not the case with applied ethnography where the goal is to collect requirements for building a new software system. Such investigations do not need to be long-term and they do not need to be continuous, either. Several carefully selected samples might be more useful for the requirements engineer's purpose.

Second, classical ethnography strives for data independence in the sense that the gathered data should remain independent from existing theoretical frameworks and classificatory systems. This is not the case with applied ethnography. Here, the requirements engineer might even bring some hypothesis to test instead of just observing the work processes.

Third, the personalization of classical ethnographies is a severe impediment for the desired objective verifiability of ethnographic studies in requirements elicitation.

Ball et al. conclude that although the majority of studies in applied ethnography violate a remarkable subset of the characteristics of classical ethnography, it can still be spoken of ethnographic methods in requirements elicitation for software systems.

2.2 Method

Unfortunately, a lot of investigations in applying ethnographic methods for requirements elicitation do not say very much about the concrete procedure. It remains unclear how the results of those studies have been acquired. Details about the things and persons to observe, the questions to ask and other important practical aspects of an ethnographic study are not mentioned.

It must be concluded therefore, that ethnographic methods for requirements elicitation consist of fieldwork in the widest sense.

2.3 Integration

So far we have looked at ethnography as an isolated method. In practice, however, it is necessary to integrate ethnographic methods into a requirements and a very broad software engineering process. Different ways of integrating ethnography into the requirements and software engineering process are possible.

First of all, it is obvious to combine ethnographic methods with other approaches for the elicitation of requirements.

Goguen [3] discusses various requirements elicitation techniques. He also addresses the question of combining several of those methods in the elicitation process. He acknowledges that none of the conventional requirements elicitation techniques are completely useless, but sees some of them more suited for certain tasks than others. The different strengths of the methods are therefore complementary. Goguen concretely proposes a zooming method for requirements elicitation. In this approach, more expensive but also more detailed methods are used to inquire problems that other, less detailed methods have found to be especially important. He remarks that the numerous methods based on ethnography are like an electron microscope: very accurate and powerful, but also quite expensive and demanding.

For example, according to the zooming method proposed by Goguen it would be possible to start with a very short ethnographic study to uncover the basic structure of a certain work area. After having received a basic overview, extended ethnographic studies in combination with detailed interviews could be used to investigate parts of the work area that have been discovered to be especially important by the first overview study.

The zooming method by Goguen is only one example of combining different approaches in requirements elicitation. Other combinations are certainly conceivable.

Yet not only a combination with other methods for eliciting requirements is thinkable. Ethnographic methods need to be integrated into the software engineering process. Hughes et al. [5] present three different approaches of integrating ethnographic methods into the software engineering process. They are outlined in the following.

Concurrent Ethnography This approach is distinguished by the fact that the ethnographic study takes place at the same time as the system is developed. Hughes et al. have practiced this approach in the development of an interface prototype for air traffic controlling. They studied an air traffic control room for a period of about four weeks. This ethnographic study was followed by a debriefing session involving both the requirements engineers and the designers. A first prototype was constructed after that. In the meantime, a further ethnographic study was conducted by the requirements engineers. Due to the nature of a prototype they did not have to write a requirements specification. A result of this study has been that the rate of utility of fieldwork for the design of a system is declining quite rapidly. Figure 1 illustrates the way concurrent ethnography works.

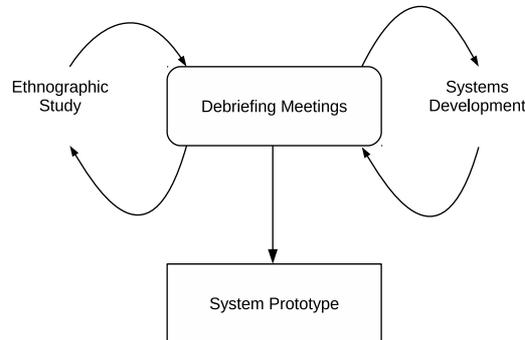


Fig. 1. Concurrent Ethnography [5]

Quick and Dirty Ethnography In this approach brief and focused ethnographic studies are undertaken to inform the developers about the environment of the system to be developed. It is accepted that the brief studies do not result in a complete and very detailed study of the workplace. According to the experiences Hughes et al. have made, this approach is able to provide “much valuable knowledge of the social organisation of work of a relatively large scale work setting in a relatively short space of time” [5, p. 7]. Figure 2 illustrates the way quick and dirty ethnography works.

Evaluative Ethnography In this case ethnography is used to validate an already existing requirements specification. This approach shows that ethnography is not only applicable in requirements elicitation but also in other areas of the requirements and software engineering process. This issue is addressed again in section 4 of this paper. Figure 3 shows the way evaluative ethnography works.

2.4 Advantages

In favor of ethnography it can be said that the use of ethnographic methods in requirements elicitation can lead to a very detailed and extensive report about a certain workplace, its users and the relationships between them. The huge amount of data collected during such a process can guarantee that there are no major omissions in the resulting product.

Ethnography also uncovers hidden details and aspects of a certain work area that would not be uncovered by different methods like interviews apart from the customer’s workplace, for example.

Already a short ethnographic study that does not need a lot of resources can result in very valuable knowledge about a certain work area.

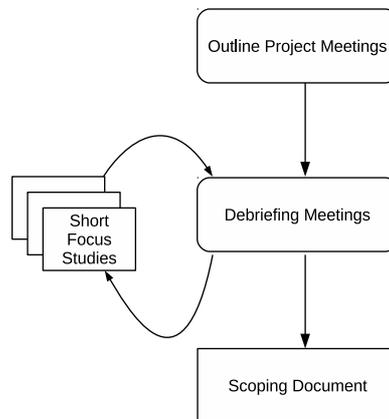


Fig. 2. Quick and Dirty Ethnography [5]

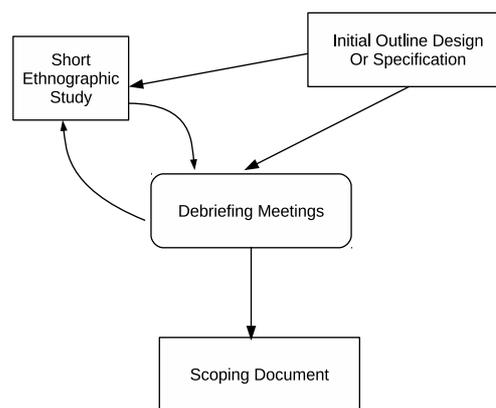


Fig. 3. Evaluative Ethnography [5]

Furthermore, ethnographic methods in requirements elicitation are covered well in the literature. It is therefore possible to select from a very broad range of methods and frameworks.

2.5 Disadvantages

One major disadvantage of ethnographic methods is that they sometimes require a great amount of time spent at the workplace of the customer. It can be a

very lengthy process which is usually not desired in the context of requirements elicitation.

Another problem is the form of the results that come from the use of ethnographic methods. An ethnographic study usually leads to detailed, lengthy, and textual descriptions [10]. However, this is not the desired form for requirements analysis and it can be very difficult to extract concrete requirements out of such data.

The already mentioned amount of results can be a problem, too. It can be very difficult and cumbersome to capture requirements out of too much data.

The data resulting from a ethnographic study is very concrete, because it is the result of intensive observations. This might be a problem if more abstract requirements should be extracted out of the study results [10].

Further, the use of ethnographic methods in requirements elicitation for software systems depends on skilled ethnographers. There is a lack of a systematic approach to conducting ethnography and this makes the method especially dependent on the ethnographer's skill [10].

As a last disadvantage the differing cultures of ethnographers and software engineers can be mentioned. This is especially relevant if the ethnographer comes with a background in the social sciences and has possibly no prior experience in requirements and software engineering [10].

2.6 Summary

Ethnography in requirements elicitation for software systems can be seen as field-work method that tries to describe a certain work area, its actors and its relationships in detail. There is no extensive guide on how to conduct an ethnographic study with the purpose of eliciting requirements available. Various approaches of integrating ethnography into the requirements and software engineering process are conceivable. The most important advantage of ethnography in requirements elicitation is that it uncovers very important aspects of a certain work area in a relatively short amount of time. The most important disadvantage is that there is no detailed guide on how to conduct an ethnographic study.

It can be concluded that ethnographic methods are useful and versatily applicable in the software engineering process. However, the application of ethnographic methods needs experienced requirements engineers and might lead to serious problems otherwise.

3 Contextual Inquiry

Contextual Inquiry is a field interviewing method that can be used for eliciting requirements for software systems. The development of Contextual Inquiry has been started in the nineties by Karen Holtzblatt. It has been further developed by Hugh Beyer and Karen Holtzblatt over the years. As a result, Contextual Inquiry is now part of a larger design approach called Contextual Design¹. However, it is

¹ The standard reference on Contextual Design is [2]. This section is largely based on that text.

still possible to look at Contextual Inquiry as isolated method for requirements elicitation.

A short introduction on Contextual Design is given in section 3.1. This is followed by a description of a typical Contextual Inquiry process in section 3.2. The integration of Contextual Inquiry into the requirements or software engineering process is discussed in section 3.3. In the sections 3.4 and 3.5 the advantages and disadvantages of this approach are presented. This section on Contextual Inquiry closes with a summary in section 3.6.

3.1 Preliminaries

Contextual Inquiry must be seen as part of the design approach called Contextual Design. Contextual Design “is an approach to defining software and hardware systems that collects multiple customer-centered techniques into an integrated design [...] process. Contextual Design makes data gathered from customers the base criteria for deciding what the system should do and how it should be structured.” [2, p. 3] So Contextual Design is a design framework with heavy emphasis on data gathered from customers. The process itself consists of several stages. In the first stage, data is gathered from the customer. It is exactly there where Contextual Inquiry comes into play.

3.2 Method

Informally, Contextual Inquiry can be summarized as follows:

“The core premise of Contextual Inquiry is very simple: go where the customer works, observe the customer as he or she works, and talk to the customer about the work. Do that, and you can’t help but gain a better understanding of your customer.” [2, p. 41]

The application of Contextual Inquiry depends like ethnography on a requirements engineer on site. The requirements engineer visits the customer at his workplace and observes his activities there. Additionally, discussions about the work done by the customer are led. This should, according to the core premise of Contextual Inquiry, result in a better understanding of the customer and his work.

So far, there seems to be no obvious and major difference from the already presented ethnographic methods.

One thing that is special about Contextual Inquiry is that it is based on the so-called apprenticeship model. The apprenticeship model is the idealized relationship model that exists between a master craftsman and an apprentice. According to Beyer et al. this is a gainful model for collecting data. Like an apprentice wants to learn from his master, a designer wants to learn from his customer about his work. And like the master, the customer is the expert who knows everything about his own work. Instead of giving the requirements engineer a list of rules on how to behave at the customer’s workplace to collect data,

he should try to enter this relationship model. As a result, the right behaviors on both sides of the relationship are installed for the purpose of learning about the customer's work. The advantage of this approach is that every requirements engineer has at least an intuitive understanding on what the relationship between a master and its apprentice is and therefore does not need to get special training on some highly theoretical background. It is easier to enter a more or less familiar relationship model than to follow a list of rules.

The Four Principles

Beyer et al. point out that the master and apprentice relationship model is fruitful but only a starting point. One important point to consider is that a requirements engineer does not want to learn about work in order to do it himself but in order to capture requirements out of it. The concrete adaptation of Contextual Inquiry is therefore based on four principles which are Context, Partnership, Interpretation and Focus. Their purpose is to concretize and correct the somewhat vague and not fully suitable apprentice and master relationship model. The four principles are explained in the following paragraphs.

Context The principle of context demands that the requirements engineer goes to the customer's workplace and observes the work there. First of all, this enables the engineer to gather ongoing experience. Ongoing experience is opposed to a summary in that it reveals all the details and hidden aspects of a certain situation. This is important because it is the purpose of contextual methods to get as close to the actual work situation as possible. Customers might give summaries of their work at workshops and in interviews apart from their workplace. However, at their workplace and while they are working the requirements engineer is confronted with the actual work. Second, the data gathered this way is concrete rather than abstract. This point is close to the one before. What the requirements engineer needs are not abstract descriptions of the work process, but rather concrete descriptions of procedures including detailed impressions. The presence of the requirements engineer at the workplace of the customer is a condition that helps to gather the desired data.

In conclusion:

“The key to getting good data is to go where the work is happening and observe it while it happens. Observing ongoing work keeps the customer concrete and keeps them from summarizing.” [2, p. 51]

Partnership The principle of partnership demands that the requirements engineer and the customer are equal collaborators in understanding the customer's work. The customer is the one who knows everything about the work he is doing. However, the introduced master and apprentice relationship model gives the master too much power, as the interviewer and interviewee model gives the interviewer – the requirements engineer in our case – too much weight. Both extremes are to be avoided. Therefore, the requirements engineer and the customer need to be collaborators.

The partnership principle should prevent a situation where the requirements engineer has a lot more power than the customer and conversely. It is important to see that a Contextual Inquiry session is not a conventional interview with a person acting as interviewer and posing the questions and a different person acting as interviewee and just answering the questions. Other relationship models that naturally occur in such situations are the expert and novice model and the guest and host model. They are both to be avoided, too.

In conclusion:

“Partnership transforms the apprenticeship relationship into a mutual relationship of shared inquiry and discovery of the customer’s work. [...] This results in an intimate relationship that allows for inquisitiveness about the details of the work.” [2, p. 56]

Interpretation The principle of interpretation demands that the data resulting from the observation needs to be interpreted. Good facts are only the starting points and not the ends of the Contextual Inquiry. Beyer et al. note that interpretation is needed to turn the fact resulting from the observation into an action that is relevant to the designer’s intent. The fact is observed during the Contextual Inquiry session and results in a hypothesis. This hypothesis then has a direct implication on the design.

Focus The principle of focus states that the interviewer needs to have a focus to see more of the work. While observing work, the requirements engineer can concentrate on very different aspects of the observed activities and all of these observations are in some way or the other true. However, the observations are only more or less relevant, depending on the purpose of the inquiry. The interviewer has a purpose to fulfill – to gather data for the creation of a new product, for example – and needs therefore to maintain focus on relevant aspects of the work.

The Interview

Contextual Inquiry as it is presented in [2] also gives the requirements engineer detailed instructions on how to lead an interview. A typical interview in Contextual Inquiry has the following structure:

1. The conventional interview: The requirements engineer and the customer need to familiarize with each other. Therefore, a conventional interaction is appropriate at the beginning of an interview. It might consist of the usual introduction of the requirements engineer and the focus of the interview so that the customer knows what the requirements engineer cares about. The engineer should introduce the master and apprentice model and make clear that the customer and his work are central for the following part of the interview. It is important to note that the result of this part of the interview is just a summary and an overview of what yet comes.
2. The transition: The requirements engineer introduces the rules for the contextual interview.

3. The contextual interview proper: The customer does his work and the requirements engineer observes and possibly interrupts the work to ask questions and to get comments of the customer. It is also the task of the engineer to lead this part of the interview. Although the customer is the expert in his own domain, the requirements engineer is the one who is the expert in interviewing the customer. Therefore he needs to pay attention to the progression of the interview.
4. The wrap-up: Finally, the requirements engineer summarizes his newly acquainted knowledge.

A whole interview session usually lasts two to three hours. It needs about 10 to 20 interviews with customers in very different roles to get all the necessary data for the analysis of the requirements.

Interestingly, Contextual Inquiry can be used to investigate a variety of different tasks. The execution of a normal task can be planned and is interruptable by the requirements engineer. This kind of task is accessible through normal interviews as introduced above. Intermittent tasks cannot be planned in advance because they happen at rare intervals. This kind of task is not that easily observable and needs additional help from the customer. Maybe he can lead a diary where he always notes whenever such a task is needed and what he does while fulfilling it. Uninterruptable tasks like surgical operations might be recorded on video or by taking detailed notes for reviewing them later together with the customer. Extremely long tasks can be analyzed by interviewing a wide range of customers playing different roles in the process at different stages of the process.

3.3 Integration

Because Contextual Inquiry is a fundamental part of Contextual Design, the integration into a larger process is trivial as long as it is Contextual Design. There seem to be no studies describing the use of Contextual Inquiry in combination with other design approaches or as isolated method within a software engineering process.

However, the fact, that a theoretical foundation for applying Contextual Inquiry with other methods and integrating it into a software engineering process that is different from Contextual Design is missing, should not obstruct the practitioner. It seems to be possible to use the very pragmatic approach even if it is just as advice on how to do the fieldwork within an ethnographic framework.

3.4 Advantages

Contextual Inquiry is a very pragmatic approach for contextual requirements elicitation. Its core premise and principles are sensible and easy to understand. The adaption of the principles in practice should not be too hard and a sometimes quite concrete guide with several smaller examples is given in [2].

An additional advantage of Contextual Inquiry is that it is part of Contextual Design. This is – as already mentioned – a design framework and provides therefore appropriate means for the further analysis and management of the data gathered by Contextual Inquiry.

3.5 Disadvantages

One disadvantage is that the use of other eliciting methods in combination with Contextual Inquiry is not explained. Additionally, it remains unclear whether Contextual Inquiry can be used as isolated method in the requirements or software engineering process.

In contrast to ethnography, there are almost no research papers on Contextual Design.

3.6 Summary

With Contextual Inquiry a further method for contextual requirements elicitation for software systems has been presented. Contextual Inquiry is part of a larger design process called Contextual Design. In its core, Contextual Inquiry is a field interviewing method. It seeks to understand the customers and their work. It is a very pragmatic and easy to understand approach. A major disadvantage is that the combination with other eliciting techniques is not discussed. Furthermore, the coverage of Contextual Inquiry in the literature is quite small when compared to ethnographic methods. In contrast to ethnographic methods, Contextual Inquiry does not explicitly stand in the tradition of classical ethnography. A major difference for the practicing requirements engineer might be that a systematic guide for Contextual Inquiry is available.

4 Discussion

In the last sections an overview on contextual requirements elicitation as well as ethnography and Contextual Inquiry has been given. After this survey it is now time to address some critical issues concerning contextual methods for requirements elicitation.

The methods for contextual requirements elicitation are quite well researched. There is a rich literature on the various techniques and methodical variations available. However, the studies underestimate the difficulties of the actual fieldwork. There is at least in the case of ethnography no systematic guide on the actual fieldwork disposable that could be used by interested requirements engineers.

What is further missing in the literature are studies on the interplay between contextual techniques and more conventional requirements elicitation methods. There are probably situations and projects where contextual methods alone are not sufficient for a successful elicitation of the requirements. Furthermore, it

should be noted that not all requirements can be elicited by the use of contextual methods. For example, legal constraints that a system needs to fulfill are very unlikely discovered by the use of contextual methods. The literature on contextual methods seems to concentrate on the elicitation of functional requirements. There are, however, also non-functional requirements that need to be captured to develop a system successfully.

As already noted, the application of the described methods does not result in a finished requirements specification. It can be difficult to extract requirements out of the wealth of the gathered data. Here again, no systematic methods are available.

Another topic where additional research would be desired is the application of contextual methods in other areas of requirements or software engineering like validation. Hughes et al. [5] have used ethnographic methods to validate an existing requirements specification. It remains unclear, however, whether contextual methods are of any use in other areas of software engineering.

Furthermore, it remains open whether the management of requirements that have been elicited by the use of contextual methods requires special frameworks and methods. Sutcliffe et al. have developed a framework called PC-RE for the requirements analysis of contextual requirements [9]. In their framework, not only requirements that meet the customer's goals, but also characteristics of the customers, and how the customers would like computer systems to achieve those personal goals can be described and managed.

So far the presentation of contextual methods in this paper has assumed that there is a manageable workplace that is limited both in respect of time and location. However, current workplaces are not always of this kind. Distributed work areas and mobile applications with rapidly changing contexts will probably become more widespread. Obviously, there is a need for research on the impact and implications of these circumstances on contextual methods as presented in this paper.

As a last point it should be mentioned that both ethnography and Contextual Inquiry require a person acting as requirements engineer at the workplace of the customer. This need is not self-evident and can be questioned. Approaches in which the customers collect the requirements themselves are at least conceivable. However, this area is at least at the moment not well researched [8].

5 Summary

In this paper an introductory overview on contextual requirements elicitation for software systems has been given. Contextual requirements elicitation has been defined as requirements elicitation that takes place at the workplace of the customer. Ethnography has been presented as a first example of a contextual method for requirements elicitation. Contextual Inquiry has been the second approach considered as exemplary method for contextual requirements elicitation in this paper. Finally, several critical remarks have been made and further research questions have been pointed out.

Contextual methods for requirements elicitation are in general well researched. The application of contextual methods results in a deeper understanding of the context in which a planned system will be used. Important aspects and hidden details that are difficult to capture with the use of more conventional elicitation techniques can be discovered with the application of contextual methods. However, the application of these methods might be difficult due to the lack of systematic guidelines.

There are several research questions concerning contextual requirements elicitation that need to be clarified. With a look to the future, the contextual elicitation of requirements for systems that have to deal with numerous and rapidly changing contexts and the contextual elicitation of requirements by the customers themselves seem to be very important and rewarding challenges.

References

1. Ball, L.J., Ormerod, T.C.: Applying ethnography in the analysis and support of expertise in engineering design. *Design Studies* 21, 403–421 (2000)
2. Beyer, H., Holtzblatt, K.: *Contextual Design. Defining Customer-Centered Systems*. Morgan Kaufmann, San Francisco (1998)
3. Goguen, J.A.: Formality and Informality in Requirements Engineering. In: *Second International Conference on Requirements Engineering*, pp. 102–108. IEEE Computer Society, Washington (1996)
4. Harper, R.H.R.: The Organisation in Ethnography. A Discussion of Ethnographic Fieldwork Programs in CSCW. *Computer Supported Cooperative Work* 9, 239–264 (2000)
5. Hughes, J., King, V., Rodden, T., Andersen, H.: Moving out from the control room: ethnography in system design. In: *Proceedings of the 1994 ACM conference on Computer supported cooperative work*, pp. 429–439. ACM, New York (1994)
6. Hughes, J., O'Brien, J., Rodden, T., Rouncefield, M., Sommerville, I.: Presenting Ethnography in the Requirements Process. In: *Second IEEE International Symposium on Requirements Engineering*, pp. 27–34. IEEE Computer Society, Washington (1995)
7. Nuseibeh, B., Easterbrook, S.: Requirements Engineering: A Roadmap. In: *Proceedings of the Conference on The Future of Software Engineering*, pp. 35–46. ACM, New York (2000)
8. Seyff, N., Graf, F., Maiden, N.: Using Mobile RE Tools to Give End-Users their Own Voice. In: *18th IEEE International Requirements Engineering Conference*, pp. 37–46. IEEE Computer Society, Washington (2010)
9. Sutcliffe, A., Fickas, S., Sohlberg, M.M.: PC-RE: a method for personal and contextual requirements engineering with some experience. *Requirements Engineering* 11, 157–173 (2006)
10. Viller, S., Sommerville, I.: Social analysis in the requirements engineering process: from ethnography to method. In: *Fourth IEEE International Symposium on Requirements Engineering*, pp. 6–13. IEEE Computer Society, Washington (1999)