Eliciting Contextual Requirements at Design Time: A Case Study

Alessia Knauss, Daniela Damian, Kurt Schneider*
Department of Computer Science, University of Victoria, Victoria BC, Canada
{alessiak, danielad}@cs.uvic.ca
*Software Engineering Group, Leibniz Universität Hannover, Hannover, Germany
kurt.schneider@inf.uni-hannover.de

***Abstract*—The need to consider context in order to understand requirements is established in requirements engineering. Recently, this has been discussed more intensively for socio-technical systems, which offer a rich spectrum of different operating contexts. Contextual requirements proved valuable to model requirements together with the context they are valid in, but there is a lack of research on how to derive them from stakeholder needs. Our goal in this paper is to explore the usefulness of existing requirements elicitation techniques for the identification of contextual requirements early, i.e. at design time. In a case study we investigate end-user viewpoints, together with interviews, scenarios, prototyping, goal-based analysis, and groupwork as a means to elicit and clarify contextual requirements already at design time. In our case study a certain combination of the applied requirements elicitation techniques stood out as most beneficial for the identification of contextual requirements. In addition, we discovered valuable indicators of differences in the operative context, for example when end-users cannot agree on refinements of specific requirements. Designers and operators of adaptive systems might benefit by taking such conflicts and resulting contextual requirements into account.**

***Index Terms*—requirements engineering; socio-technical systems; adaptive systems; exploratory; design time**

I. Introduction

Socio-technical systems consist of people, their social relationships, and underlying technical solutions. Not only do requirements for the technical (software) parts of these socio-technical systems frequently change, but the system operating environment and therefore the context in which certain requirements are valid change as well. In such a situation, it is important to investigate how the technical solution continues to support the processes as well as the social relationships of the stakeholders in a socio-technical system. It is not enough to focus on requirements, but also the specific context of a requirement needs to be taken into account. *Contextual requirements* promise to offer a suitable mechanism for the analysis of socio-technical systems [7] and the changes in their operating environments. They are defined as requirements that are only valid in a specific context and combine a description of the requirement and the context [1], [13].

Yet, while the usefulness of contextual requirements has been discussed in literature, there is little research on how to analyze, or more specifically to use existing requirements engineering (RE) practices for their elicitation at *design time* or *runtime*.

In this paper, we present a case study with the goal of *exploring the usefulness of existing requirements elicitation techniques in identifying contextual requirements at design time*. We had the opportunity to work with the Human Resources Department at the University of Victoria in revising its job applicant tracking system. Reports of actual RE practice in real projects is rare in the literature [17]. In-depth studies of RE practice are difficult to carry out given the human, organizational and political aspects that surround software projects. Yet, they are very important in providing insights about the application of RE techniques in the identification and analysis of requirements in practice.

To explore the application of existing elicitation techniques to identify contextual requirements in the software project at the University of Victoria, the first author acted as the requirements analyst in the project. The applicant tracking system was to be replaced to better serve the needs of thousands of stakeholders. We applied different requirements elicitation techniques, e.g., interviews, prototyping, scenarios, goal-based approaches, and focus groups to this real world socio-technical system.

We were able to document a number of contextual requirements when applying particular requirements elicitation techniques in a particular order: First we identified requirements through interviews and focus groups, attempted to understand the rationale behind them through interviews, then used prototyping to get a detailed understanding of these requirements in context. Next we identified conflicts between different end-users when discussing requirements in detail together in focus groups. These discussions helped identify the need for contextual requirements. Finally we identified, through interviews with the respective end-users, the different context related to requirements so that we could document the contextual requirements.

The insights from our report of a case study of applying requirements elicitation techniques to identify contextual requirements include:

- Conflicts among stakeholders indicate the need for contextual requirements: Requirements of several stakeholders are not alternatives and are not consistent with each other - they rather apply under certain conditions

that are determined by the context.
- Viewpoints are valuable to identify such context related to requirements and to analyze contextual requirements.
- Prototypes are particularly helpful to understand the context of conflicting requirements in detail.

In our case study we also discuss the usefulness of identifying contextual requirements in the design of adaptive systems. In our case the need for system's runtime adaptation (by identifying contextual requirements) became apparent based on given end-user requirements and their complex operating environments. The context part offered a starting point to analyze triggers for adaptation, while the requirement part defined the adaptation goal. Designing such a system as an adaptive system would allow to fulfill otherwise conflicting requirements of different stakeholders by adapting to their context specific needs.

The paper is structured as follows: In Section II we motivate our research and define some concepts for the case study. In Section III we present our case study on elicitation of contextual requirements. Section IV contains our experiences and insights that we gained in our investigation. Finally, in Section V we discuss threats to validity and conclude the paper in Section VI.

II. Motivation and Research Gap

A. Contextual Requirements: Definition and Example

To investigate elicitation of contextual requirements we use the context definition of Dey: *Context* is "any information that can be used to characterize the situation of an entity. An entity is a person, place, or object that is considered relevant to the interaction between a user and an application, including the user and applications themselves." [9]

Motivated by Inverardi and Ali [1], [13] we define *contextual requirements* as requirements that are only valid in a specific context, linked to context-attributes that define this specific context. *Context-attributes* are defined as a specific type of context information that is relevant.

An example of elicitation of contextual requirements from our case of the applicant tracking system:

1) *Requirement elicited:* Consider the requirement (R1) *the system should be able to screen the applicants in case the recruiters have too many applications for one position* and its rationale that screening helps with identifying qualified applicants.
2) *Relevant context identified:* A relevant context-attribute in this case is the number of applicants that applied to an open job posting and can be small (e.g. 15) or high (e.g. 100). Another context is the department in which the job is posted, the qualification of applicants etc.
3) *Resulting contextual requirement (CR1):* If the number of applicants is high, the screening functionality should propose the 15 most qualified candidates. If the number of applicants is small, the functionality depends on the department. In some departments, recruiters need to be able to get more information from the few candidates that applied. In this case chat functionality could enable recruiters to connect to the applicants and receive more information or to clarify information already submitted.

Depending on the design of the system under development it could be implemented as a traditional expert system with if then statements. Another option could be to implement algorithms for example to classify the qualification of applicants into the system. Additionally, feedback loops [3] could be implemented so that the system can "learn" how to best classify the qualification of applicants as well as for the evolution of the identified contextual requirement CR1. Feedback loops could also be used to identify situations in which contextual requirements are outdated, for example due to changed end-user needs or changed operational context. Through such kind of feedback loops the system can adjust its own behaviour at runtime and therefore would resolve in a self-adaptive system [3].

B. Vision: RE for Self-Adaptive Systems

For adaptive systems requirements have to be identified by a requirements analyst as always, but in addition it has to be understood when the system should adapt. Then, at runtime, the system monitors the context and decides how to adapt to the current context based on the information defined [4], [1]. Supporting elicitation methods should therefore help to understand when the applicability of a requirement depends on context.

Self-adaptive systems are aware of their requirements and able to execute basic RE activities on their own at runtime to change their (adaptive) behaviour [23], [18]. As contextual requirements depend on runtime context they are prone to change and points for system self-adaptation. The elicitation of contextual requirements already at design time allows to implement the knowledge about points of adaptation into the system. At runtime the self-adaptive system can monitor and modify such requirements through the integration of feedback loops [26].

C. Research Gap: Req. Elicitation for Adaptive Systems

Current research on adaptive systems concentrates mainly on modeling, architecture and monitoring, see for example [24], [1], [28], [21]. However, research previews present the importance of further investigations in RE for adaptive systems, e.g. [5], [20], [28]. Especially defining the context as well as which context-attributes might change at runtime as part of the RE activity at design time, that links requirements with context is an important activity that needs further investigation [5], [6].

D. End-Users and Viewpoints as a Source for Contextual Requirements

End-user involvement in RE is crucial for systems that integrate context-awareness [25], [22]. Van der Zanden argues that software prototypes of the developed system would

be the best way to validate context-awareness [25]. His work is only based on literature and is missing empirical investigation. We use mock-ups from an existing system and similar systems to set users in context, so that they can imagine how the system under development can look like.

Dalpiaz et al. [7], Lapouchnian et al. [15], and Ali et al. [2] investigate requirements analysis for adaptive socio-technical systems. They model and analyze contextual requirements that represent end-user needs in goal models. An interplay of modeling and analysis as well as research that investigates the communication around contextual requirements is necessary. As one of the few real case studies we extend their work by investigating the communication about requirements, so that this information can then be modelled. In turn, models are used as a foundation for communication about requirements in our investigation.

In the preparation of our case study we identified the importance of different user viewpoints. Therefore, we assumed that different user viewpoints have an impact on the requirements for our system. We use the *viewpoint* definition by Easterbrook: "A viewpoint is a self-consistent description of an area of knowledge with an identifiable originator" and is "the context in which a role is performed" [11], [12]. The choice of requirements elicitation techniques for our study is based on the usefulness of the technique to be used in combination with viewpoints approaches. Zowghi and Coulin present techniques and approaches for requirements elicitation [29]. Additionally, they show techniques that can be used complementary: Interviews, groupwork, prototyping and goal-based analysis are the complementary techniques to viewpoints.

III. Case Study on Eliciting Contextual Requirements at Design Time

This section gives an overview of our case study, presents the system under investigation and the requirements elicitation techniques we used in our exploration for the elicitation of contextual requirements.

A. Research Methodology

TABLE I: Requirements elicitation activities in the project. Second iteration describes the case study we report on in this paper.

Iteration	Resposibility of	Purpose	Techniques used
1	Professional Req. Analyst	Requirements elicitation	Interviews, focus groups
2	First author	Contextual requirements elicitation & prioritization	Interviews (3+24) in combination with: scenarios, prototyping, goal-based approaches. Focus groups (7).

According to recommendations of Rubå and Cruzes [10] to study requirements elicitation processes based on field work and give contextualization, we decided to conduct an exploratory case study [16] with the office of Human Resources of the University of Victoria, Canada. Our aim for this case study is to explore and continually reflect on the usefulness of different requirements elicitation techniques (alone or in combination) in the elicitation of contextual requirements. A case study of exploratory nature [19] allows us to observe natural behaviour of stakeholders in socio-technical systems and the results of elicitation techniques. Such end-user behaviour and the outcome of different requirements elicitation techniques cannot be observed from the outside, but needs to be studied in context. Instead, a researcher has to stay a longer time in the field and become a part of local culture [16].

Our research group was asked to help with the RE process for the applicant tracking system at the University of Victoria at a time when some initial requirements have already been elicited but some dissatisfaction with the elicited information was expressed. We decided to integrate one researcher as requirements analyst into this project.

Table I shows the process of RE activities in the entire project. At the start of our investigation, the first iteration of requirements analysis had already been completed by a professional requirements analyst and integrated all end-user groups. At this point, the RE process was considered to be complete, with the prioritization of requirements missing. Our task (second iteration of requirements activities in this project) was to revisit the participants of the initial RE activities to enable prioritization of the documented requirements as well as ensure that all end-user requirements were covered.

For research purposes we were interested which of the elicited requirements depend on context and how this context interplays with the socio-technical system. Our task was to combine different existing requirements elicitation techniques for the elicitation of contextual requirements and related context-attributes that define the context. Systems integrating contextual requirements have to capture this context at runtime to decide about necessary adaptation. In the scope of this study we investigate how the dependence of requirements on context and the specific context itself can be determined based on discussing requirements with end-users. Specific context-taxonomies for different kinds of systems are already proposed in literature (e.g. [27]), which can be used for identification. Nevertheless, the problem that we encountered is that there cannot be a single specific taxonomy for context that applies to all situations, as context-attributes change based on the project itself, the conditions of the project etc.

Hence, our goal in this investigation is to explore existing requirements elicitation techniques with the goal to combine both, elicitation of requirements as well as the identification of related context that the requirements depend on at the same time at design time through the involvement of end-users. We used existing requirements elicitation techniques that are complementary to viewpoints: interviews, prototyping, goal-based analysis, and groupwork [29]. We add

scenarios as a technique to be able to talk about requirements with users and that gives us the possibility to understand requirements in context of other requirements, which is important in a complex setting. We explored the usefulness of the chosen requirements elicitation techniques for the identification of contextual requirements based on the need that we felt in the interview.

Before presenting our insights in the next section, we describe the system under investigation, users of the system, and requirements elicitation techniques we applied in more detail.

B. System under Investigation: Applicant Tracking System

Our study is based on a real project that aims at upgrading the staff and faculty applicant tracking system with thousands of users and introduce it into a bigger systems landscape. In the past the existing system was only used for hiring staff (library assistants, nurses, janitorial, etc.). In the replacement process a new group of end-users is introduced additionally to staff – the faculty staff members. Another goal for the replacement is to integrate the applicant tracking system in the complex environment of a socio-technical system – the University of Victoria system landscape as well as the thousands of users that use (parts of) this landscape – by integrating different systems that are used together. In the past the transition between the applicant tracking system and other systems was based on the manual work of end-users (e.g. printing documents and typing them in the other systems) who transferred the data into related systems. For example, the replacement system should facilitate to push or pull applicant data to and from a separate administrative system that is used to manage personal and accounting data for all employees.

For the RE effort, thousands end-users of the system were classified into 6 large groups of end-users whose requirements are considered in the development of the system. The end-users and their goals are as follows:

- **Human resources** staff has different roles and goals that range from helping other end-users in solving their problems with the system to checking that regulations are adhered to collective agreement rules.
- **Finance** staff is interested mainly in confirming the availability of budget for new positions (also known as position control). Payroll/HRIS processes job and employee information for new hires.
- **Recruiters for staff** from different departments are looking to fill positions.
- **Recruiters for faculty** from different departments are looking to fill positions.
- **Operational unit** staff supports recruiters in the recruitment process.
- Finally, the biggest group consists of **applicants** that are interested in being hired and who use the system during the application process.

The case study we present offers highly diversified end-user groups. Some groups were substantially larger than others. The groups include people with a high level of responsibility that are quite busy and have little time to participate in the RE process, power-users that are using the system for hiring employees several times a week, novice users that have only used the system once, and users with other possible combinations of use in between these ranges.

Only a small sample size of each group could be considered in our study. Each sample covered 4-6 participants for each of the six core groups during the requirements elicitation activities. For some of these groups, the participants of the case study were only a very small sample size, whereas for the smallest group these 4-6 participants covered almost the entire end-user group.

C. Requirements Elicitation Techniques

In this section we describe the requirements elicitation techniques and how they were used in detail.

Interviews: Interviews with end-users constitute the elicitation technique most commonly used in practice [4]. This was also the preferred technique for our project. We started with a series of three interviews with end-users from different groups for the purpose of understanding the domain, identifying sources of requirements, analyzing the stakeholders and selecting techniques and approaches [29]. The results of some of these activities were already well prepared and documented by the professional requirements analyst, so that this effort was minimized for us consisted of studying the documents.

Following, we conducted 24 additional interviews for our research purpose of identifying contextual requirements. We used voice recording for later analysis of meetings with end-users. We interviewed each end-user for approximately one hour. We enriched the interviews with task-analysis, scenario walkthroughs, and prototyping in different combinations. Goal-based approaches were used for requirements analysis in parallel. In some of the interviews we used all three techniques. In these interviews we started with discussions about the end-user goals that the system should support and the tasks they would be performing while using or aiming to use the new system. Following this, we asked the end-users to describe their scenarios. The last task was to talk about their (most important) requirements. This question sometimes helped end-users to think out of the box, and not only to think about their scenarios. Additionally, by talking about the most important requirements we got a feeling of what was perceived by the end-users as important.

Scenarios: The requirements documentation that we received from the professional requirements analyst did not include scenarios. The requirements were only mapped to high level user goals, e.g., "interviewing" or "applying for a job posting". During many of the interviews, we used scenarios for discovering requirements. Scenarios allow talking to end-users about their requirements step by step and to understand the goals that end-users aim to achieve through the completion of tasks.

We asked the end-users for their scenarios that the replacement system should cover. If the end-user had difficulties to come up with scenarios, we divided this task in three subtasks. The end-user should
(i) Describe the scenarios for using the existing system,
(ii) how they could be improved, and
(iii) which of the steps or even which scenarios can be added beyond the scope of the existing system.

We give a high level example of a scenario with the goal "fill a new position" that was discussed with one of the end-users. Note that this scenario can be divided into subgoals/tasks.

1) Looks at the application of each applicant.
2) Review applications.
3) Forward them to the hiring committee.
4) Create a short list (which is done by the hiring committee).
5) Email applicants that are shortlisted.
6) Telephone or email applicants (choice is on personal preference).
7) Schedule interview appointments and document information concerning the applicant in the system.
8) Put information into the system in the case when someone has been chosen.

After an applicant has been chosen to fill a position, a new entry with the employee data needs to be created in a separate administrative system. In this system all employees have an account, containing personal data, accounting data, etc. Both of these systems can be integrated in a way that data can be securely pushed and pulled between this system and new applicant tracking system. There are some workflows that may need to bridge the two systems, e.g., accounting approvals.

Prototyping: We explored the use of the following prototypes as visual representations to discuss requirements: 1) Screenshots from the existing system, 2) screenshots from another very successful system that fulfilled the same purpose, and 3) screenshots that were proposed by end-users.

Screenshots of the actual system and also other systems that contained further features were used for clarifying requirements. In some interviews end-users talked about their experience with other systems which fulfilled the same purpose. They explained some of their requirements based on their experience. We asked them to show us the system. We used these then in some of the following interviews.

Focus Groups: After discussing the requirements with a few end-users from the same group (the minimum was three interviews) we set up two hours focus groups. In total, seven focus groups with 3-7 participants were conducted to discuss the priorities of requirements. We employed a simple process: If end-users from the same end-user group did not give the same priority rating to a requirement it was discussed.

We based our prioritization process on the work of Karlsson et al. and used the group sorting technique for prioritizing requirements with stakeholders [14]. At the end of our prioritization process, user requirements fell into three different groups of requirements: Requirements with an average of high, medium and low priority. The group with medium priority requirements contained many requirements with different ratings (high, medium and low), which we call here conflicting requirements. These requirements were analyzed in more detail, to identify the reason for the contradicting end-user needs.

Goal-based requirements analysis was used in parallel to document and analyze end-user goals as well as their tasks to fulfill these goals. We used task analysis in the beginning of most interviews in order to understand commonalities and compare the viewpoints to other end-users.

IV. Insights from the Application of Elicitation Techniques to Identify Context. Requirements

In the following we present our insights from the case study. One main insight is the need for a combination of different requirements elicitation techniques for the elicitation of contextual requirements. We present the combination that helped us elicit contextual requirements in our case study.

End-users from the same end-user group did not necessarily agree on requirements and priorities due to different user viewpoints. We identified the importance of viewpoints to identify context related to requirements early on in our investigation. One of the initial interviews was especially important for understanding the dynamics in the project. The person we interviewed was responsible for the help desk and knew the problems of the existing system as well as the needs of the different end-user groups. Through this interview we understood that project success would depend on the consideration of viewpoints in the prioritization of requirements. This prioritization was most challenging because of the complexity of the end-user groups. The scenarios for the six groups were partially heterogeneous and end-users from one group could not see the importance of the other groups, because of the limited understanding for the viewpoints of others. Therefore we considered different viewpoints as a source for requirements.

Requirements elicitation techniques should not assume that stakeholders have similar operating contexts. While analyzing the documented requirements that resulted from the first iteration we found 33 requirements with an average rating for high priority, 86 for medium priority, and 48 for low priority. From the 48 with low priority, there were even 22 requirements with low priority for the target group, from which the requirement originated. We investigated those 22 requirements out of 167 in more detail. We made an observation that requirements were important only for some end-users from the target group. They seem to have been documented when the requirements analyst encountered and understood them for the first time. In our case when we tried to understand those requirements in the interviews with the stakeholders the requirement was not investigated in detail, because the analyst, misled by her previous knowledge, failed to recognize the subtle differences that were caused by

different end-user context. Prioritization would not help in this case as different end-user viewpoints are not considered for requirements. This is one of the main causes for the problems we encountered with prioritization. Asking users cannot solve this problem, because usually end-users do not have other end-users but only their own viewpoint. They think they are understood and cannot know that a similar requirement with potentially contradicting details already exists.

Designing a system as an adaptive system through the consideration of contextual requirements helped resolving conflicting requirements.

In our case study the indication that adaptation to context is necessary was given by the fact that no agreement could be achieved with respect to prioritization of some of the requirements (especially requirements with an average rating of medium that contained high, medium as well as low individual ratings) among end-users without considering the specific context for requirements.

One indicator for the need of systems adaptation to context is the fact that the execution of a requirement does not only depend on the input from the end-user the system provides the functionality to, but also from other sources (e.g., for CR it is the number of applicants, type of position, qualification of applicants). These sources are the context-attributes that need to be considered for context-adaptation. At runtime, when identifying these context-attributes, the system should decide which functionality it offers to each end-user based on the exact value for the attribute. At design time this context-awareness and therefore consideration of context of the end-users for adaptation needs to be investigated. One opportunity for the identification is talking about different context with end-users and creating a shared understanding. Conflicts are great sources for contextual requirements and important context-attributes.

The original requirements elicitation based on established techniques was not complete and satisfying from the perspective of the project manager. The first indicator for this was the lack of prioritization of the requirements. Closer investigation on this issue revealed several problems:

- Different end-user groups did not agree on priorities. For this reason, the priorities of each group were weighted based on the importance of the group.
- Contradicting priorities of requirements even within groups: It was not possible to achieve an agreement on the priorities of requirements, not even within end-user groups, i.e. two end-users with the same role would often disagree on priorities.
- Contradicting requirements: End-users would agree on an abstract requirement, but would offer contradicting specific requirements when asked for details. Often, these detailed requirements would exclude each other.

Having finer grained end-user groups would not have been a solution, because the prioritization was planned on the given level of abstraction, i.e. based on end-user roles. Even with finer grained groups, an agreement about the prioritization and contextual requirements would have needed to be achieved. Finer grained groups only shift the problem up and require focus groups that consist of representatives from different end-user groups which would further increase the effort. Also, it was not possible to create finer grained groups in our case, because any two end-users would agree with one requirement and disagree with another requirement.

Conflicts among end-users helped resolve different viewpoints and identify important contextual requirements.

We focused on combining elicitation techniques with viewpoints. The need for this combination became apparent during a particular conflict in a focus group meeting. After one third of the scheduled time, one end-user left the room, apparently angry about the situation ("It was not a pleasure."). We have conducted a follow-up interview in which we used prototyping and were able to understand the end-user viewpoint as well as identify context-attributes for the discussed contextual requirement. This interview revealed that the discussion during the focus group was not aligned to the views of this end-user, even though all participants belonged to the same end-user group, had the same role, and no differences were visible. The end-user could not find herself in the discussion results. In order to achieve a consistent requirements description, it is important to be aware that conflicts can lead to great results and resolve them in a less frustrating way, by facilitating them. Otherwise, the support of important end-users might be lost during elicitation and development, rendering the resulting system useless.

During our study, the following steps proved useful in our requirements elicitation approach:

1) Confront each end-user with a specific requirement in context.
2) If two end-users disagree, analyze their viewpoint and the context they are preassuming.
3) Add as many context-attributes as needed to fulfill their needs through contextual requirements.

From the requirements perspective, this algorithm leads to important context-attributes for systems adaptation, i.e. the ones that determine when the system needs to adapt in order to fulfill the end-users' requirements.

Prototypes were useful in understanding the differences of end-users in context. When the first attempt to prioritize the requirements failed, our assumption was that the initial requirements elicitation was not sufficiently detailed. Therefore, we applied prototyping to get more detailed requirements from end-users [8]. For adaptive systems the activity of identifying and capturing the context for the development of the software system plays an important role. Therefore prototyping might have a really important role in identifying contextual requirements. Therefore our hypothesis was that prototyping would unveil enough context to identify important contextual requirements and thus overcome the limitations of only using one requirements

elicitation technique. However, prototyping alone was not sufficient to overcome these limitations and to deliver consistent prioritization and requirements without contradictions. Our endeavour revealed that the initial requirements were complete and actually in a good shape. Prototyping lead to the same requirements, and simply added more details.

Prioritization was still not possible because of contradicting requirements, and no agreement could be achieved between different end-user groups, or even among the end-users of one group. The major problem was the wide range of end-users and their different needs (same end-user group, same requirement). It seemed that the documented requirements were only documented from the viewpoint of one end-user, probably the first who had communicated this requirement. While the other end-users from the same group generally had the same requirement, when discussed in detail, they were, in fact, slightly different, demonstrating the need for different system functionality.

In our project, prototyping helped to identify these differences. By using the same prototypes when talking to end-users about the same requirement, an understanding of the differences in the needed functionality could be reached in most of the cases. In a few cases, the personal preferences were what made the difference.

No single elicitation technique was sufficient to capture contextual requirements on its own. To summarize we propose the following seven-step approach for a similar process that someone wants to use (see Table II for details). In this project none of the existing elicitation techniques was successful in eliciting contextual requirements on its own. This approach was successful because it allowed end-users agree on prioritization and resolve inconsistencies in viewpoints:

1) Identify all requirements from the (sample of) users (which in our case study this activity was done by a professional requirements analyst).
2) Understand the rationale behind these requirements, to prepare for their discussion with other end-users.
3) Use appropriate prototypes to better understand the needs of end-users in context.
4) Let end-users prioritize the elicited requirements.
5) Identify conflicting requirements or prioritization of requirements.
6) Understand the rationales behind conflicting requirements. Are there cases where the end-users can agree on prioritization if documenting the requirement as contextual requirement?
7) If steps 1 to 6 do not lead to a successful understanding of the requirements then conduct further interviews with end-users who had contradicting priorities for the same requirement.

This process was successful because it led to contextual requirements which allow end-users to agree on prioritization. The discussion about different viewpoints helped end-users to understand the specific context of other end-users. This allowed the disagreeing end-users to accept that in a specific context, a given requirement is important. Discussing the complete picture during focus groups, especially different context that the end-users of the same user group had with respect to one specific requirement helped to establish a complete picture of the adaptation needs concerning this requirement.

TABLE II: Following these steps we could identify contextual requirements at design time.

#	Technique	Description	Explanation
1	Interview, Focus Groups	Identify requirements	Having a complete list of requirements is necessary for a complete understanding.
2	Interviews	Rational behind priorities	Understanding why something is important helps to prepare negotiation.
3	Prototyping	Detailed understanding	Establish reference for negotiation: Details for req. are important to be able to compare this req. to similar from other end-users.
4	Focus Groups	Prioritization of req.	
5		Identify conflicting req.	These conflicts between end-user req.are sources of important context-attributes.
6		Understand priorities establish common ground on detailed req.	How do those detailed req. differ? Derive context-attributes and related contextual req.
7	Interviews	Understand priorities and why they differ. Back to step 2	Probably information is missing for a complete understanding.

V. Threats to Validity

This exploratory case study is based on authentic data that were analyzed and interpreted by the first author. The main concern here is that one of the authors was responsible for requirements elicitation. Associated with this there might be some limitations of our ability to objectively discuss the results. Firstly, it is hard to objectively judge if there were faults in the requirements elicitation techniques as applied by the first author. Secondly, the goal of this exploratory study was to investigate the use of requirements elicitation techniques for the elicitation of contextual requirements. With this goal in mind, the first author might have introduced a bias into the results. However, we are not reporting quantitative results, but share qualitative results in our example from this specific perspective.

As a single case study, our insights might have a low generalizability to other projects. Our insights are at least applicable to other applicant tracking systems for universities. Many issues we encountered seem to result from a wide range of end-users covered in this study. This indicates that a significant amount of our experiences are more generally applicable to the development of today's complex systems that often face the similar challenge of having to cover a wide range of end-users in the RE process.

VI. Conclusion

In this paper we share our insights from an exploratory case study on contextual requirements elicitation. Specifically, we investigated the usefulness of existing requirements elicitation techniques in identifying contextual requirements at design time. Our insights from this study include that it was possible in our case to derive the need of a system to be adaptive from early end-user requirements, i.e. because certain types of conflicts are encountered between different end-users concerning particular requirements that were due to different user viewpoints.

Based on this insight, we are able to draw an important connection between a) stakeholder conflicts as can be discovered by applying viewpoints in requirements elicitation and b) adaptive systems. Not only are contextual requirements a valuable input to the design and running of adaptive systems. More important, designing the system as adaptive allows to fulfill requirements that are in conflict with each other. The system can adapt to the context of its stakeholders and offer the applicable requirement. Further, contextual requirements could be a starting point for the implementation of self-adaptive systems when implementing separation of concerns and feedback loops, e.g., [26]. Our insights could be valuable for practitioners in supporting the development of adaptive systems with a large number of end-users.

Further research is needed to transfer our insights into a more general approach that is applicable to other systems as well as domains for the elicitation of contextual requirements. Additionally, it would be interesting how this work can be applied and extended to cases when an existing running adaptive system needs manual identification of context related to requirements. The conflicts that surface based on different viewpoints might be useful for adaptive systems in general – a potential correlation that should be investigated in future work.

Acknowledgements

Special thanks goes to the project managers for giving us the opportunity to study requirements elicitation in a real project and the participants of the study for their time and support. We thank the SEGAL and Rigi group from University of Victoria, as well as Eric Knauss for their feedback on drafts of this paper. This research was funded by NSERC NECSIS Grant.

References

[1] R. Ali, F. Dalpiaz, and P. Giorgini. A goal-based framework for contextual requirements modeling and analysis. *Requirements Engineering Journal*, 15(4):439–458, 2010.

[2] R. Ali and A. Griggio et al. Optimizing Monitoring Requirements in Self-adaptive Systems. In *BPMDS 2012 and EMMSAD 2012*, 362–377, 2012.

[3] Y. Brun, G.D.M. Serugendo, and et al. Engineering Self-Adaptive Systems through Feedback Loops. In *Self-Adaptive Systems*, 48–70, 2009.

[4] V. Castelli, R. Bertone, P. Thomas, and A. Oliveros. A Requirements Engineering Process extended to Context Information Management. *IEEE Fifth International Conference on Research Challenges in Information Science*, 1–6, May 2011.

[5] B. H. Cheng and R. Lemos et al. Software Engineering for Self-Adaptive Systems: A Research Roadmap. In *Software Engineering for Self-Adaptive Systems*, 1–26. Springer-Verlag, 2009.

[6] A. Classen and A. Hubaux et al. Modelling Variability in Self-Adaptive Systems: Towards a Research Agenda. *Int. Workshop on Modularization, Composition and Generative Techniques for Product-Line Engineering*, 1(2):19–26, 2008.

[7] F. Dalpiaz, P. Giorgini, and J. Mylopoulos. Adaptive socio-technical systems: a requirements-based approach. *Requirements Engineering Journal*, 18(1):1–24, September 2011.

[8] A. M. Davis. Operational Prototyping: A New Development Approach. *IEEE Software*, 9(5):70–78.

[9] A. K. Dey and G. D. Abowd. Towards a Better Understanding of Context and Context-Awareness. In *1st Int. Symposium on Handheld and Ubiquitous Computing*, HUC '99, 304–307, 1999.

[10] T. Dybåand D. S. Cruzes. Process Research in Requirements Elicitation. In *IEEE Int. Workshop on Empirical Requirements Engineering (EmpiRE)*, number 7465, 36–39, 2013.

[11] S. Easterbrook. Handling conflict between domain descriptions with computer-supported negotiation. *Knowledge Acquisition: An International Journal*, 3:255–289, 1991.

[12] S. Easterbrook. Domain Modelling with Hierarchies of Alternative Viewpoints. In *IEEE 1st Int. Symposium on Requirements Engineering*, number January, 65–72, 1993.

[13] P. Inverardi and M. Mori. Requirements Models at Run-time to Support Consistent System Evolutions. In *IEEE Int. Works. on Requirements@Run.Time*, 1–8, 2011.

[14] J. Karlsson, C. Wohlin, and B. Regnell. An evaluation of methods for prioritizing software requirements. *Information and Software Technology*, 39(14-15):939–947, January 1998.

[15] A. Lapouchnian and J. Mylopoulos. Modeling Domain Variability in Requirements Engineering with Contexts. In *28th Int. Conference on Conceptual Modeling*, 115 – 130, 2009.

[16] T. C. Lethbridge, S. E. Sim, and J. Singer. Studying software engineers: Data collection techniques for software field studies. *Empirical Software Engineering*, 10:311–341, 2005.

[17] N. A. M. Maiden. Exactly How Are Requirements Written? *IEEE Software*, 29(1):26–27, 2012.

[18] N. Qureshi and A. Perini. Requirements Engineering for Adaptive Service Based Applications. In *IEEE Int. Requirements Engineering Conf. (RE'10)*, 108–111, Sep. 2010.

[19] Per Runeson and Martin Höst. Guidelines for conducting and reporting case study research in software engineering. *Empirical Software Engineering*, 14(2):131–164, December 2008.

[20] M. Salehie and L. Tahvildari. Self-Adaptive Software: Landscape and Research Challenges. *ACM Transactions on Autonomous and Adaptive Systems*, V(N):1–40, 2009.

[21] M. Salifu, Y. Yu, and B. Nuseibeh. Specifying Monitoring and Switching Problems in Context. In *IEEE 15th Int. Conference on Requirements Engineering*, 211–220, Delhi, India, 2007. IEEE Computer Society.

[22] N. Sánchez-Pi, J. Carbó, and J. M. Molina. A knowledge-based system approach for a context-aware system. *Knowledge-Based Systems*, 27:1–17, March 2012.

[23] P. Sawyer and N. Bencomo et al. Requirements-Aware Systems: A research agenda for RE for Self-adaptive Systems. In *IEEE Int. Requirements Engineering Conf. (RE'10)*, 95–103, Sep. 2010.

[24] Alistair Sutcliffe and Pete Sawyer. Modeling Personalized Adaptive Systems. In *CAiSE 2013*, 178–192, 2013.

[25] G. van der Zanden. Requirements Engineering for Context-Aware applications. In *Twente Student Conference on IT*, 2008.

[26] N. M. Villegas and G. Tamura et al. DYNAMICO: A Reference Model for Governing Control Objectives and Context Relevance in Self-Adaptive Software Systems. *Software Engineering for Self-Adaptive Systems II - Lecure Notes in Computer Science*, 7475:265–293, 2013.

[27] N. M. Villegas and H. A. Müller. Managing Dynamic Context to Optimize Smart Interactions and Services. In M. Chignell, editor, *The Smart Internet*, 289–318. Springer Berlin Heidelberg, 2010.

[28] D. Weyns and M. U. Iftikhar et al. Claims and Supporting Evidence for Self-Adaptive Systems: A Literature Study. In *Software Engineering for Adaptive and Self-Managing Systems*, 89–98, 2012.

[29] D. Zowghi and C. Coulin. Requirements Elicitation: A Survey of Techniques, Approaches, and Tools. In *Engineering and Managing Software Requirements*, 19–46. 2005.