Requirements-driven Social Adaptation: Expert Survey

Malik Almaliki[1], Funmilade Faniyi[2], Rami Bahsoon[2], Keith Phalp[1] and Raian Ali[1]

[1] Bournemouth University, UK
[2] University of Birmingham, UK

Abstract. [Context and motivation] Self-adaptation empowers systems with the capability to meet stakeholders' requirements in a dynamic environment. Such systems autonomously monitor changes and events which drive adaptation decisions at runtime. Social Adaptation is a recent kind of requirements-driven adaptation which enables users to give a runtime feedback on the success and quality of a system's configurations in reaching their requirements. The system analyses users' feedback, infers their collective judgement and then uses it to shape its adaptation decisions. **[Question/problem]** However, there is still a lack of engineering mechanisms to guarantee a correct conduction of Social Adaptation. **[Principal ideas/results]** In this paper, we conduct a two-phase Expert Survey to identify core benefits, domain areas and challenges for Social Adaptation. **[Contribution]** Our findings provide practitioners and researchers in adaptive systems engineering with insights on this emerging role of users, or the crowd, and stimulate future research to solve the open problems in this area.

Keywords: Requirements Engineering, Adaptive Systems, Social Adaptation

1 Introduction

In self-adaptive software community there has been a great deal of emphasis on architectures to support design and development of adaptation, models for anticipating and reacting to changes in the managed system and methods for verifying properties of these systems [1,2]. Ultimately, self-adaptivity is a meta-computing capability which enables a system to reason about itself and its dynamic environment so that it can formulate the right decisions to reach stakeholders' requirements [3].

While success on these foundational fronts has contributed significantly to the field, the role of users in the adaptation process has only recently become a main focus. This can be partly attributed to lessons learnt from successfully deployed self-adaptive systems such as Rainbow [4], where it was found that the adaptation process was not transparent to users. An example of such transparency limitations can be illustrated by the insufficient explanation offered by self-adaptive system about why a course of actions was chosen instead of alternative actions to meet the users' requirements.

Early research in self-adaptive systems limited users' ability to steer adaptation with the good intention of maximizing system autonomy and minimizing human efforts. However, this would lead to adaptation decisions that were valid but only temporarily since users were not given a voice in the iterative validation of these decisions after software was deployed [5]. Consequently, one of the identified research challenges in the engineering of self-adaptive software systems road map is:

[To devise a way of] *“analysing feedback types from human-computer interaction and devising novel mechanisms for exposing the control loops to the users, keeping the users of self-adapting systems in the loop to ensure their trust”* [1].

Although the role of users in the adaptation process has recently been recognized [5,6,7,8], there is still a lack of consensus and holistic approaches on how to engage the users and the crowd in that process. In this paper, we address this problem and conduct an expert survey to gather and analyse the knowledge of experts in adaptive systems research. We give the acquisition of users’ feedback a special focus due to its vital role in enabling this kind of adaptation. Our survey provides practitioners and researchers in self-adaptive systems with insights and challenges to consider when involving users, individually or as a crowd, in the adaptation process.

The paper is structured as follows. In Section 2 we briefly discuss Social Adaptation. In Section 3 we describe the study’s objectives and design. In Section 4 and Section 5 we present the results of the first and second phase of the survey, respectively. We discuss threats to validity in Section 6 and conclude the paper in Section 7.

2 Social Adaptation

Social Adaptation is defined as a system’s autonomous ability to analyse users’ feedback and choose an alternative behaviour which is collectively shown to be the best for meeting requirements in a context [5]. Social Adaptation claims to have the benefit of improving the transparency of the self-adaptive system and raising users’ trust in it, since users are treated as first-class entities in both the engineering and also the operation of such systems. In fact, over time of using the software, users may be able to shape the decision-making process in a way that can only be done by today’s experts.

Some researchers have pursued similar visions under themes such as requirement-aware self-adaptive systems [9], requirement monitoring at run-time [3], and social adaptation in pervasive software systems [6]. All these efforts adhere to a notion of representing users’ requirements or trust relationship among users (in [6]) as run-time objects that can be used by the system to reason about the adaptation process. Other researchers use the term of socially-adaptive software differently to refer to software agents which are socially adaptive in the sense of their ability to comply to social norms, e.g. [10]. Social Adaptation, as described in [5], is unique in the sense that instead of catering to the requirement of a user or subset of users at run-time, it harnesses the “wisdom of the crowd” to adapt the system in a way that is deemed best by end-users’ collective judgement rather than the decisions of an elite group of users or those of developers. To put it another way, Social Adaptation pursues the goal of a democratic-like, consensus-based social approach to adapting software systems to meet users’ requirements.

In Social Adaptation, users act as monitors and provide software with information via their feedback. This introduces a range of challenges for the engineering of this human-based monitor. Reviewing the literature, we could not identify systematic approaches for feedback acquisition at runtime. The impact of users’ feedback and how users behave when providing feedback is still ambiguous as discussed in [11].

The lack of engineering processes for feedback acquisition would lead to poorly designed feedback collection mechanisms and this could harm the quality of collected feedback, users' experience and the quality of adaptation and evolution decisions [7]. Owing to its importance, our Expert Survey will give a particular focus on the engineering challenges of feedback acquisition in Social Adaptation.

3 Expert Survey Design

The study's objectives were to poll the opinion of experts on (i) the principles and primitives for enabling Social Adaptation (ii) the role of users' feedback in steering software adaptation, and (iii) the engineering of software-based feedback acquisition.

3.1 Experts Selection

Experts selection can have a high effect on the survey outcomes and the acceptability of the result in the wider community [12]. Since we are tackling a multidisciplinary research area and in order to have a diversity of viewpoints, we targeted experts from Requirement Engineering and Adaptive Systems research community with additional focus on at least one different related domain: HCI, Human Factors in Computing, Psychology, Privacy and Security Engineering, Socio-Technical Systems Engineering and Social Computing. Our inclusion criteria allowed for experienced participants who are knowledgeable in their respective fields, evidenced by proven publication track record. Although the majority of our experts work in academia, they either worked in industry previously or were engaged in collaborative projects involving industrial partners. To make sure these participants had sufficient experience and knowledge about the discussed issue, some assessment questions regarding their knowledge and experience were asked at the beginning of the questionnaire.

According to expert elicitation practitioners, the number of experts to be included should be at least six, otherwise we would not be confident about the quality of the conclusions and their generalizability [13]. In the first phase of our survey 35 experts were invited; 29 forms were returned. Considering the average actual time taken to complete the survey (35 minutes), the size of the form and the amount of effort required to complete it, we consider this as a good response rate. In the second phase we invited the 29 experts who participated in the first phase and only 21 forms were returned. We invited additional 5 experts so that we got a total of 26 forms completed.

3.2 Design, Test and Distribution of the Survey

We used online questionnaires as a data collection method for our study because our experts were widely distributed geographically (five countries). The questionnaire contained both types of questions: open-ended questions and close-ended questions. The open-ended questions were used to ensure that we minimize the risk of missing significant information and to give participants a space to include information they felt was relevant. Closed questions were employed to ensure that we get a high response rate and to put less effort on participants when answering the questionnaire [14].

Survey questions were deduced and extracted from two talks, given by two of the authors, followed by a brainstorming session on Requirements-driven Social Adaptation. The sessions took place on March 2013 as a part of a project meeting, which included academics in the computing departments of three universities. The participants set included 12 researchers who have a variety of relevant expertise including Requirements Engineering, Self-adaptive Systems, Dynamic Software Product Lines, Cloud Computing, Machine Learning and Human Factors in Computing.

The questions focused on the value and benefits of Social Adaptation for both developers and clients, its application areas, whether it has to be autonomous or semi-autonomous and its technical development challenges. A good part of the discussion focused on the acquisition of users' feedback, how to engineer it, and whether it should be adaptive as well. The survey script contained 25 questions discussing and investigating these points.

Questionnaires need to be tested on typical respondents before the actual data collection stage begins to ensure their readiness and clarity [15]. Our questionnaire was tested first on three respondents who met our inclusion criteria. After the test and revision, experts were sent an email containing a brief description of our purpose of the survey and asking them to participate in the Expert Survey. We gave a period of two weeks for them to come back to us with their input. Surprisingly, the response rate was high (29 out of 35) which is an indicator that the field is relevant and timely especially to Requirements Engineering which is a primary research area of our surveyed experts.

4 First Phase Results

The returned survey forms were analysed and responses were cleaned up and irrelevant/inconsistent answers were excluded. A descriptive analysis on the quantitative part of the survey was conducted to describe the data and to get the feel of it. A qualitative analysis was applied to the open-ended questions of the survey which included coding the response and creating categories to identify patterns and trends in the responses.

4.1 Social Adaptation Benefits and Value

Social Adaptation claims to offer valuable benefits for both developers and users. This claim raises important questions that need to be addressed by experts. The following 4 questions attempt to dig a little deeper, that is to understand, not only to what extent Social Adaptation is beneficial but also to understand better the nature and context of those benefits among different groups. The questions also vary in their focus. In brief, Q1 to 3 consider benefits for developers and clients or users with Q4 attempting to consider areas that are either particularly fruitful or, in contrast do not offer particular benefits.

Q1: How would you rate the benefits of Social Adaptation: (a) For software developers? (b) For software clients.

Beneficiary / Rating	High	Medium	Low
Developer	13	14	2
Client	20	8	1

A rating of *Low* implies Social Adaptation is not beneficial; *Medium* implies that there are benefits but not necessarily significant; significant benefits are rated as *High*. There is a consensus among experts that Social Adaptation, if realised, is a useful concept to developers (93% chose medium/high) and clients (96% chose medium/high). The higher perceived benefit to clients is perhaps not surprising, as users will have more active role in steering the adaptation process.

Q2: What are the benefits of Social Adaptation for software developers?

Social Adaptation, as indicated by experts' responses, offers valuable benefits for both developers and users of adaptive software. **[Finding 2.1]** Acquired knowledge through users' feedback can be used to build and refine models used by the system or to improve the accuracy of reactive or predictive adaptive algorithms for various aspects of the self-adaptive system. New knowledge may also reveal latent requirements that were not known before. Developers of self-adaptive systems can therefore use Social Adaptation to: (1) improve problem resolution tactics by identifying bugs and scenarios that cause software crashes and poor performance, (2) better prioritise requirements and maximise the productivity of limited development resources, (3) identify the distribution of software use across age groups, geo-location, time of day etc., and (4) build knowledge-bases of contextual profiles, which are hard to elicit at design time where the users have not used the system in real settings yet.

In contrast to Q1, where the numbers are revealing; this open question gave us a great deal of insightful comments from the expert survey. In terms of benefits, respondents noted that Social Adaptation: *"Provides insights from the user perspective to software developers on aspects of the system that need to change."* (EX24). *"Learning about and adapting to new (or un-elicited) requirements and making software more aware of new contexts seamlessly."* (EX25). *"Up-to-date knowledge - accessible unobservable knowledge - able to react to new events (in a faster way) - more knowledge shared knowledge"* (EX1). *"Considering adaptation early avoids making hard and expensive changes afterwards when the system is running."* (EX12).

[Finding 2.2] Future socially-driven adaptive systems may disrupt the current development paradigm of self-adaptive systems, in terms of time to market or deploy, by reducing the upfront effort in design phase to the barest minimum. By taking the socialized view of adaptation, the system will only provide a platform for users to express their preferences, whilst design decisions are collectively made by users at run-time. This indeed makes Social Adaptation different from other approaches (e.g. Agile software development) in which the variability points of the software and execution environment will be learnt at run-time based on feedback provided by users which makes. This is a realistic assessment since users of today's software system vary widely in their preferences (perhaps, influenced by culture, norms, age group, location etc). Designers of adaptive systems will focus on engineering open, configurable, and extensible platforms, instead of debating functionality choices.

Q3: What are the benefits of Social Adaptation for users?

The ultimate goal of Social Adaptation is to satisfy users' requirements efficiently by enabling users to steer and tailor the adaptation process. Our experts agreed that Social Adaptation was of most benefit to users. This correlates with responses to the first question. **[Finding 3.1]** The benefits cited included improved trust (users feel their voice is considered), **[Finding 3.2]** user satisfaction (software behaves according to users' judgement), **[Finding 3.3]** transparency (adaptation decisions is visible to users), and **[Finding 3.4]** confidence in self-adaptive system *"1- Acknowledgement of clients' opinion 2 Visible involvement of clients in the adaptation"* (EX22), *" Having a software with an adaptive behaviour based on similar users / past experience and participation / involvement in a community is very gratifying to many users"* (EX5).

The data also revealed that the involvement of users and their ability to collectively configure the software on the fly at run-time could result in users perceiving the software as a partner rather than a tool *"Clients will have more user-friendly software as a result of the analysis performed on information received from them. Their confidence and trust in the adaptive systems may grow. They will be able to provide focused and real-time feedback to the developers which can in return empower them."* (EX3). Indeed this is consistent with the overarching objective of self-adaptation, where the software is expected to adapt to users' needs and not the other way around. Since Social Adaptation is a group-based adaptation, the software will be able to quickly adapt to group norms and beliefs, without the conventional maintenance-evolution phases.

Experts also indicated that users' involvement should be accommodated under some constraints and users should not be always involved in shaping and validating software adaptation. Restricting users' involvement is due to a negative effect on users' experience (e.g. annoyance), which might arise when involving them too much *"if every system would ask often to confirm something, the end-users would be overwhelmed and they would not react at all"* (EX1).

Aside from the apparent benefits of Social Adaptation to users (as listed above), one unusual finding is the perceived impact on the software life-cycle of self-adaptive software systems. The idea of fully automated user-driven evolution or reduced human involvement in the evolution process already raises many challenges. An example of these challenges could be the way users' feedback is being collected and analysed by the system. The findings and challenges in this area will be discussed in another question.

Q4: Can you nominate certain areas and application domains where Social Adaptation: (a) has distinguished benefits (b) should not be used?

A common theme raised by our respondents was the need for a user-centred or human-oriented software and user bases in which preferences of the entire user population collectively steer the adaptation. We classify the identified application areas as follows:

- **[Finding 4.1] Mobility intensive systems** where software is used in different contexts, e.g., driving navigation systems *"Telecom industry will be highly interested in such applications, Content intensive applications"* (EX10).
- **[[Finding 4.2] Large-scale systems** such as SaaS clouds, where the software is in global demand and developers are unable to elicit preferences of groups of users distributed around the world *"Any software system that has a very large community of users, e.g., smartphone applications."* (EX7).
- **[Finding 4.3] Real time management systems** where crowd-sourcing will empower the system monitor and enhance the decision making. For example, evacua-

tion scenarios or congestion management at train stations or airports *"- unobservable areas - areas with lots of traffic and different end-users and needs (airport, central train stations, shopping malls) - mobile devices (apps with unknown end users)"* (EX1).

- **[Finding 4.4] Highly interactive systems:** These are software that is frequently used for variety of purposes where it is hard to know a priori how users will judge quality in the diverse contexts of use and human-computer interaction. Mobile, pervasive, and social networking applications fall into this category of highly interactive systems. *"Systems with repetitive tasks. The driving navigation system is a good example. Perhaps an operating system, adapts to common usage. Also embedded electronics like refrigerators, heating, etc. could adapt to behaviour without criticality."* (EX14).
- **[Finding 4.5] Prototyping tools** in moderately dynamic systems, where feedback from Social Adaptation can be used to infer user needs before a final implementation is carried out. Here, there is an assumption that the rate of adaptation is relatively controllable by human development effort after the final system is deployed *"in prototyping for requirements engineering activities possibly, to find out which model to focus on in the actual implementation"* (EX16)

Our experts considered Social Adaptation inapplicable in the following domains:

- **[Finding 4.6] Critical systems** where wrong Social Adaptation could result in disaster or huge financial loses.
- **[Finding 4.7] Security sensitive applications**.
- **[Finding 4.8] Non- or less-interactive systems** under the control of centralised authorities such as payroll system or an embedded system. A comment from an expert in regard to areas outside the scoop of Social Adaptation *"Safety Critical Systems where real-time data input may results in disasters e.g. a nuclear power plant. High Secure Systems - that is sealed/closed systems e.g. military missile systems."* (EX3).

It is interesting to uncover the subtle difference between "personalised" adaptation and "social" adaptation as evident in candidate application areas listed above. The former refers to a type of user-driven adaptation where the objective is to meet the requirement of user(s) with mutual non-conflicting preferences. Crucially, some users may choose not to conform to popular opinion, therefore, they should be given the freedom to deviate from the choice deemed best by the group (e.g. for some privacy reasons). On the other hand, social adaptation is a different concept as the preferences of the entire user base (including conflicting ones) is collectively used by the system to adapt in a way deemed best for the group. Figure 1 illustrates the difference between these concepts. Applying Social Adaptation in the previous domains is a promising opportunity to empower adaptation quality. The reason is that the potential to get a wide range and large volume of users' feedback is high and that the users' feedback is meaningful as the interaction between users and the software is intensive.

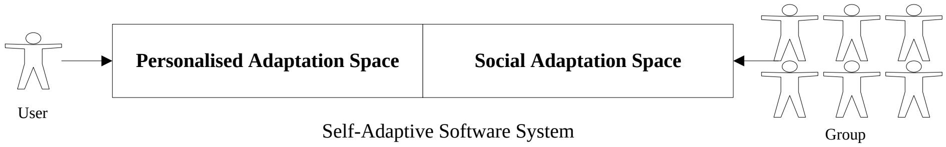

Fig. 1. Personalised Adaptation Versus Social Adaptation in Self-Adaptive Software

4.2 Challenges to Supporting Benefits to Developers and Users

Utilising on-line Learning: The role of on-line learning is a key in realising open platforms for socially-driven adaptation since there is a question of: how do we build a system when little is known about the users of the system. Existing work in the self-adaptation literature mainly uses learning-based approaches to model decision-making about computational configurations at run-time (e.g. [16,17]). Unlike these efforts, the role of learning here is that of learning user trends, behaviours, and adaptively resolving conflicts among user preferences at run-time. Since user behaviour and preferences are not static, i.e. users themselves evolve, a fully open social-adaptation platform should empower users to decide the protocols and resolution tactics for their collaboration. Consequently, **[Challenge 1]** there is the problem of identifying what learning models to use for enabling user interaction and conflict resolution at run-time and **[Challenge 2]** how the process of mining users' feedbacks should be conducted to inform recommendations which are consistent with the system's requirements.

Gauging User Involvement: The challenges include **[Challenge 3]** how to measure users' involvement with the system as a main descriptor of their feedback, **[Challenge 4]** identifying the degree to which users are allowed to configure the software on the fly at run-time and **[Challenge 5]** and specifying restrictions for their involvement (e.g. users can provide feedback after a certain period of use).

Monitoring Adaptation Spaces: From Figure 1, it can be observed that while a user is able to independently tune parameters in his/her personalised adaptation space, the system's social adaptation space, which also affects the user, is a product of a collective configuration. **[Challenge 6]** The research challenge here is to develop models and languages to allow users to specify their preference on the way their requirements are reached (e.g. when to rely on the crowd and when to take their personal choices). Another challenge **[Challenge 7]** is to develop mechanisms to allow software to deduce such users' preference without getting them explicitly involved.

4.3 Implementation Choices (Autonomy and Feedback Acquisition)

The following questions dig much deeper and attempt to look at issues to consider in implementing Social Adaptation, with Q5 examining autonomy and Qs 6-8 focussing on aspects and developments challenges of users' feedback acquisition process since it plays a significant role in enabling Social Adaptation.

Q5: Knowing that relying on Social Adaptation is a user's choice, are there cases where software should still ask users to confirm its adaptation decision?

The degree of autonomy in socially-adaptive systems has been always debated. Should the system take an autonomous full control on the adaptation process? Or should the user interfere sometimes? We extract, analyse and discuss experts' opinion in regard to autonomy in Social Adaptation.

96% of the respondents agree that user confirmation is essential before an adaptation action autonomously suggested by software is allowed to impact the system. The remaining 4% claim that adaptation actions should not require user confirmation since this is why self-adaptive systems are autonomous *"User should be not too much bothered. Moreover I think that user identification should be automatic in the mechanism adopted to get user feedback."* (EX20).

Even though the consensus tilts towards user involvement in confirming adaptation actions, many experts believe that the answer to this question is not a binary (yes or no). In many cases, the choice of whether user should be involved depends on the type of services provided by the adaptive system (e.g. its criticality), the implication of the action on user's privacy, security, and financial spending. Also, "nice to have" autonomous social adaptation actions should not outweigh core functionalities of the system, hence users need to choose what is important to them in each context.

Q6: Can the quality of collected feedback be affected by the way it is collected?

All respondents agreed that the way feedback is collected has an effect on the quality of the feedback. The exact implication of the collection mechanism can be manifested in terms of:

- **[Finding 6.1] Time:** Asking users for feedback when they are busy may result to poor responses or it may be discarded. Finding the right time to ask for feedback is important *"Asking in a busy moment of the end-user will result in only yes/no answers. Asking if the end-user is bored will result maybe in creative feedback, but not necessarily high quality."* (EX1).
- **[Finding 6.2] User interface:** Short, concise, clear questions should be preferred to long, complicated ones. The user interface should allow users to express their biases using wider band of options, such as Likert scale [18], rather than conventional yes or no options.
- **Finding 6.3] Language:** The phrasing of the question, for example based on the language proficiency of the user, will determine how users interpret and respond to questions. Even for experienced users, misinterpretation is sometimes inevitable due to the ambiguity inherent in natural languages *"simple interfaces are essential, avoid long texts or complicated questions. Things such as "like" and "dislike" may be especially effective"* (EX5).
- **[Finding 6.4] Quality of users:** This involves asking the right user population for feedback and ensuring the size of users is representative of the group characteristics *"Is it the right user group? How representative is the feedback? User profiles? Can you capture user's perception or viewpoint of the question asked? How is the feedback question phrased? Any domain specific language?"* (EX3).
- **[Finding 6.5] User's mood** Some interesting responses suggest strongly that the mood of users should be factored into the feedback acquisition process. While the mood of users may impact the quality of feedback, it is still hard to monitor the

emotional state of users (e.g. happy, bored, excited, angry etc.) during feedback acquisition. Perhaps advances in recognizing emotions through facial expression [19] could be helpful in this area.

Q7: Is the development feedback acquisition mechansims technically challenging?

Experts stress that developing software-based feedback acquisition poses a variety of technical challenges due to changing context of use and users' evolvement. Almost two-thirds (65%) of the responses indicated that the users selection and interaction style is a key challenge which includes incentivising users to give feedback, when to ask for feedback, who to ask for feedback, how to interact with users without annoying them and the usability degree when giving feedback *"Uncertainty, building a user-friendly interface, convincing the user of the importance of it."*(EX7). *"The design should incentivize impatient and ignorant users to give feedback. Implementation would not be the problem, designing is the main challenge."*(EX18).

Responses also indicated that, engineering of users' feedback acquisition is a multidisciplinary process and it has a potential usefulness and strong relationships with various domains such as, Requirement Engineering, Ubiquitous systems, HCI, Context-aware systems Social Science, Psychology, Recommendation Systems and Machine learning.

From the various experts' answers to the engineering challenges of a software-based feedback acquisition we can deduce that the engineering of an adaptive software-based feedback acquisition stands out as a technically challenging process. The first step of gathering feedback already raises many questions: What type of feedback should be asked of users? How should feedback be cleansed, represented, processed, filtered, and selectively adopted for use by the systems? If users are relied upon to steer the adaptation process, then the system must be equipped with capabilities to cope with the ambiguity flaws of natural languages.

Additionally and in relation to the findings of the previous question (Q6), the extracted engineering challenges of the feedback acquisition can also be considered as factors that can affect negatively/positively the quality of collected feedback. Addressing these challenges can possibly improve the quality of collected feedback. For example, developing an incentive scheme for users might improve the quality of their given feedback.

Q8: If feedback acquisition is adaptive, what could be the adaptation drivers?

Context is important for the choice of feedback acquisition methods, and experts agreed that an adaptive feedback acquisition mechanism is a necessary enabler to decide ways of acquiring feedback. Some possible drivers for such adaptive mechanism suggested were:

- **[Finding 8.1] User experience:** E.g. usage frequency. This could inform how often users should be asked for feedback. For example, a less frequent user may find providing feedback meaningless, since they hardly use the software in the first place *"Usage information of user's laptop (or other smart devices). E.g. if a user is browsing web sites or watching a video, probably he/she is free."* (EX17).

- **[Finding 8.2] Application constraints:** Such as the application model, domain model, and level of interactivity of the software are likely to influence ways of acquiring feedback *"this should include several components, including a user model, application model, domain model, and a general feedback or adaptation model."* (EX8).
- **[Finding 8.3] Direct enquiry:** Involves asking the users if they wish to provide feedback, if yes, how often they wish to do so and what methods they would like to use for providing such feedbacks *"Ask the user what they prefer. When is the best time to give feedback, what form would they like?"* (EX14).

  The identified trend here is that drivers of adaptive feedback acquisition should not be studied in isolation. Such drivers may trade-off against each other. A user that provides feedback frequently, for example, will only find answering the direct enquiry questions useful, as a way of improving his/her feedback provision.

4.4 Research Challenges to Implementing Social Adaptation

Degree of Autonomy: It is interesting to observe that although experts advocate that Social Adaptation is useful for meeting users' requirements, autonomously, based on the crowd feedback, they still believe individual users should be in the loop during the decision-making process of their software. **[Challenge 8]** It raises the question of how much control users are willing to surrender to software systems. For example, in modern autopilot assistant systems, pilots take a supervisory role while software controls the flight of airplanes. The challenge here seems to be psychological in nature, since users are happy to trust the system when they are involved in the decision-making. Does this mean users trust their own socially-generated decisions less than expert knowledge encoded in systems such as Auto-pilot? Suppose, users were able to collaboratively fly an aircraft, would it land safely? Perhaps, this trustworthiness issue is why experts believe that Social Adaptation should not be used in critical systems but in less critical systems (See Q4).

Impact of Collection Approach and Importance of Mood: Investigation into the impact of user mood on the quality of feedback in specific application domains may require evidences from psychology *"This is mainly a psychological issue, finding the right time and modality and give incentives to the user for providing a good feedback"* (EX5). Advances in neuroinformatics could be helpful in this area. Some experts suggest that feedback should only be requested for features that are frequently used by the user. **[Challenge 9]** This will require mechanisms for monitoring user's feature usage statistics/trends and using these results to inform which feedbacks are requested from the user.

[Challenge 10] Some domain-specific feedback acquisition languages and mechanisms might be needed. Some feedback mechanisms may work better in some application areas than others. This challenge is akin to problem in requirement elicitation based on application areas and user experience. Perhaps some lessons can be learnt from the requirements community to address this challenge.

[Challenge 11] Additionally, we could turn to mature fields like HCI to learn how interfaces are built to gather feedback from users in a variety of contexts or even to use innovative features such as voice-based feedback acquisition rather than purely text which might make the process easier and more enjoyable for users *"[users] will provide more feedbacks to a system that can support voice recognition than others without this feature."* (EX17).

Impact of User Selection and Interaction Styles: In a software-based feedback acquisition, a further important challenge is catering for the users selection and interaction style. More specifically, there are challenges in the following aspects **[Challenges 12-18]**: 1) modelling users styles (including incentives, 2) deciding when to ask for feedback, 3) deciding what type of feedback to ask for, 4) deciding with whom to interact, 5) deciding how to interact and avoid annoying or confusing users, 6) deciding how to design for maximized usability in feedback acquisition and 7) deciding how to ensure trust and reliability of acquired feedback.

Feedback Acquisition Drivers: [Challenge 19] The challenge here is indeed the need to identify the relevant drivers of the adaptive acquisition of users feedback and **[Challenge 20]** engineer these drivers in a way that is non-intrusive to users. In addition, Social Adaptation is applicable and useful in various domains and the availability of a systematic approach for engineering an adaptive users' feedback acquisition is highly valuable. It could bring promising benefits for users and developers in the different domain where adaptation is recommended and, perhaps, different disciplines like marketing and e-commerce. **[Challenge 21]** Therefore, the development of an application-independent framework for an adaptive users' feedback acquisition is also a key challenge of users' feedback acquisition.

5 Second Phase Results

From our Expert Survey responses, we were able to deduce and extract a set of core findings and challenges in the area of Social Adaptation and engineering of users' feedback acquisition. In order to confirm the set of extracted findings and the degree of relevance and difficulty of our extracted challenges to the Requirements Engineering research community, we conducted a second phase survey. We invited the 29 experts who responded in the first phase and 21 forms were returned. Then we invited 5 new experts who attended at least one of the seminars given by one of the authors on Social Adaptation. They all responded which made a total of 26 completed form in this phase.

The survey was designed and delivered following our approach in designing the previous Expert Survey (see Section 3). Before experts answer the survey, they were given a brief reminder about the purpose of our first-phase Expert Survey and then a brief description about the second-phase survey and the purpose of it. In addition, a brief description before each set of challenges was given to highlight why it was extracted/identified as a challenge to give a clearer vision to experts before answering the questions.

The questions were developed to discuss and gather experts' opinion in regard to the following three points:

- Confirming our findings of the first phase. We focused on the debatable findings which did not receive a high percentage of consensuses in the first phase. We marked the findings using the tag **[Finding x.y]** in Section 4. We gave three options for each finding: *Agree*, *Partially Agree*, and *Disagree*.

- Measuring the degree of challenge in each of the extracted challenges (we marked the challenges using the tag **[Challenge x.y]** in Section 4). We gave the following three options: *[Ch: A]*: It is challenging and it requires significantly new approaches, *[Ch: B]*: It is challenging but it can still be solved by extending and customizing existing approaches. *[Ch: C]*: It is not really challenging and solutions already exist in the literature.
- Measuring the relevance degree of each challenge to the area of Requirement Engineering (RE). We gave the experts three options here: *[RE: A]*: It is very relevant to RE research. *[RE: B]*: It is not strictly relevant to RE research, but having a solution for it is still beneficial to RE. *[RE: C]*: The challenge and solution are not relevant to RE research and practice.

The following tables present a summary of our second survey findings.

Finding	**Agree**	**Partially**	**Disagree**
[2.1]	50%	46%	4%
[2.1]	**69%**	27%	4%
[3.1]	65%	34%	4%
[3.2]	**73%**	27%	0%
[3.3]	38%	58%	4%
[3.4]	58%	31%	11%
[4.1]	**69%**	23%	8%
[4.2]	50%	38%	12%
[4.3]	46%	46%	8%
[4.4]	**85%**	15%	0%
[4.5]	65%	27%	8%
[4.6]	**81%**	15%	4%
[4.7]	34%	54%	12%
[4.8]	50%	38%	12%
[6.1]	**92%**	8%	0%
[6.2]	**85%**	15%	0%
[6.3]	**88%**	12%	0%
[6.4]	**77%**	19%	4%
[6.5]	**69%**	31%	0%
[8.1]	**81%**	19%	0%
[8.2]	65%	30%	4%
[8.3]	50%	46%	4%

Table 1. The confirmation of experts on the findings of the first phase

In Table 2, a high degree of challenge is given to engineering challenges related to enabling users to steer the adaptation process and the degree in which they are willing to steer it (e.g. challenge 6 and 8). This high degree of challenge is perhaps due to the lack of models and languages for enabling users to express their adaptation preferences and the lack of studies on the degree of autonomy in socially-adaptive systems. Another noticeable high degree of challenge was given to challenges related to engineering feedback acquisition for different application areas and empowering adaptivity in it (e.g. challenge 10, 19 and 20). The reason behind this high degree of challenge could be the obvious lack of systematic approaches for engineering feedback acquisition.

In addition, challenges related to users' involvement, feedback collection and interaction styles and feedback mining to inform adaptations show a high degree of relevance to RE (e.g. challenge, 2, 5, 6, 10, 13, 14 and 16). This high degree is perhaps because experts believe that users' involvement in the adaptation process, ability to provide feedback in their preferable way and the system's ability to react to their feedback accordingly is a user' requirement that should be systematically engineered and efficiently met.

Challenges	[Ch:A]	[Ch:B]	[Ch:C]		[RE:A]	[RE:B]	[RE:C]
[1]	46%	46%	7%		50%	42%	8%
[2]	15%	77%	7%		**65%**	27%	8%
[3]	38%	54%	7%		42%	46%	12%
[4]	27%	57%	15%		50%	42%	8%
[5]	15%	65%	19%		**61%**	35%	4%
[6]	**61%**	34%	7%		**84%**	11%	4%
[7]	50%	50%	0%		50%	42%	8%
[8]	**58%**	34%	7%		65%	27%	7%
[9]	15%	77%	7%		57%	27%	15%
[10]	**61%**	35%	4%		**69%**	31%	0%
[11]	15%	54%	30%		27%	57%	15%
[12]	42%	50%	8%		46%	34%	19%
[13]	35%	50%	15%		**65%**	27%	7%
[14]	19%	58%	23%		**62%**	34%	4%
[15]	24%	50%	8%		50%	42%	8%
[16]	31%	61%	8%		**65%**	23%	12%
[17]	35%	46%	19%		54%	38%	8%
[18]	35%	57%	7%		58%	35%	7%
[19]	**58%**	42%	0%		46%	50%	4%
[20]	**54%**	46%	0%		54%	42%	4%
[21]	46%	46%	8%		50%	46%	4%

Table 2. The challenge degree and the relevance to RE of each of the challenges of the first phase

6 Threats to Validity

Our expert survey has three main threats to validity:

- The first threat is one of the common issues when designing a questionnaire and relates to ensure whether the questions were understood by all experts as intended. This threat is somehow addressed as we conducted a pilot test on typical respondents then some questions were revised and modified to ensure clarity. This was done for both phases of our survey.
- The second relates to the low percentage of our experts who have industrial experience in adaptive systems. The reason is that adaptive systems are not yet widely applied in industry and much of the work is still in academia. This could mean that our results are flavoured with more judgements coming from academia than industry.
- The third relates to the fact that Social Adaptation is a forward-looking way of developing adaptive systems. This would mean that the answers of our experts are fairly speculative. However, given that most of the elements of this domain as well as the survey questions are directly related to the main areas of expertise of our experts (e.g. requirements engineering, adaptive systems, HCI, and social computing) we would consider that the answers are good enough to draw credible insights.

7 Conclusion

This paper has synthesized findings from a two-phase Expert Survey of 29 experts in the first phase and 26 experts in the second phase on the topic of Social Adaptation and the challenges posed by the mechanisms for collecting user feedback, to steer the adaptation process. The consensus among experts is that Social Adaptation is a highly beneficial concept to both developers and clients of self-adaptive systems. However, enabling Social Adaptation is a technically challenging process due to the lack of models and mechanisms for enabling such a concept. Engineering approaches are highly needed for Social Adaptation to empower users' involvement in shaping adaptation decisions and to systematically develop the feedback collection process and interaction styles as well as feedback mining. The paper has highlighted research challenges in the areas of providing an enabling platform for Social Adaptation and the design of adaptive feedback acquisition mechanisms that fits user context.

Acknowledgement

The research was supported by an FP7 Marie Curie CIG grant (the SOCIAD Project) and by Bournemouth University through the Fusion Investment Fund (the BBB and the VolaComp projects) and the Graduate School PGR Development Fund. We would also like to thank Sarah Williams for insights on conducting qualitative research.

References

1. Cheng, B.H.: Software engineering for self-adaptive systems. Springer-Verlag, Berlin, Heidelberg (2009) 1–26
2. Oreizy, P., Gorlick, M.M., Taylor, R.N., Heimhigner, D., Johnson, G., Medvidovic, N., Quilici, A., Rosenblum, D.S., Wolf, A.L.: An architecture-based approach to self-adaptive software. Intelligent Systems and Their Applications, IEEE **14**(3) (1999) 54–62
3. Fickas, S., Feather, M.S.: Requirements monitoring in dynamic environments. In: Requirements Engineering, 1995., Proceedings of the Second IEEE International Symposium on, IEEE (1995) 140–147
4. Cheng, S.W., Huang, A.C., Garlan, D., Schmerl, B., Steenkiste, P.: Rainbow: Architecture-based self-adaptation with reusable infrastructure. In: International Conference on Autonomic Computing, IEEE (2004) 276–277
5. Ali, R., Solis, C., Omoronyia, I., Salehie, M., Nuseibeh, B.: Social adaptation: when software gives users a voice. In: ENASE'12: 7th International Conference Evaluation of Novel Approaches to Software Engineering. (2012)
6. Esfahani, N., Malek, S.: Social computing networks: a new paradigm for engineering self-adaptive pervasive software systems. In: Proceedings of the 32nd ACM/IEEE International Conference on Software Engineering-Volume 2, ACM (2010) 159–162
7. Pagano, D., Brügge, B.: User involvement in software evolution practice: A case study. In: Proceedings of the 2013 International Conference on Software Engineering. ICSE '13, Piscataway, NJ, USA, IEEE Press (2013) 953–962
8. Ali, R., Solis, C., Salehie, M., Omoronyia, I., Nuseibeh, B., Maalej, W.: Social sensing: When users become monitors. In: Proceedings of the 19th ACM SIGSOFT Symposium and the 13th European Conference on Foundations of Software Engineering. ESEC/FSE '11, New York, NY, USA, ACM (2011) 476–479

9. Sawyer, P., Bencomo, N., Whittle, J., Letier, E., Finkelstein, A.: Requirements-aware systems: A research agenda for re for self-adaptive systems. In: The 18th IEEE International Requirements Engineering Conference (RE), IEEE (2010) 95–103
10. Van Riemsdijk, B.: Socially adaptive software. Awareness Magazine. (2013)
11. Pagano, D., Maalej, W.: User feedback in the appstore: An empirical study. In: the 21st IEEE InternationalRequirements Engineering Conference (RE). (2013) 125–134
12. Linstone, H.A., Turoff, M.: The delphi method. Addison-Wesley Reading, MA (1975)
13. Cooke, R.M., Probst, K.N.: Highlights of the expert judgment policy symposium and technical workshop. Resources for the Future Washington, DC (2006)
14. Leung, W.C.: How to design a questionnaire. student BMJ **9**(11) (2001) 187–189
15. Franklin, S., Walker, C.: Survey methods and practices. Statistics Canada, Social Survey Methods Division (2003)
16. Elkhodary, A., Esfahani, N., Malek, S.: Fusion: a framework for engineering self-tuning self-adaptive software systems. In: Proceedings of the eighteenth ACM SIGSOFT international symposium on Foundations of software engineering, ACM (2010) 7–16
17. Tesauro, G., Jong, N.K., Das, R., Bennani, M.N.: A hybrid reinforcement learning approach to autonomic resource allocation. In: The IEEE International Conference on Autonomic Computing, ICAC'06., IEEE (2006) 65–73
18. Likert, R.: A technique for the measurement of attitudes. Archives of psychology (1932)
19. Adolphs, R.: Recognizing emotion from facial expressions: Psychological and neurological mechanisms. Behavioral and cognitive neuroscience reviews **1**(1) (2002) 21–62