Contextual Requirements Prioritization and Its Application to Smart Homes

Estefanía Serral[1(✉)], Paolo Sernani[2], Aldo Franco Dragoni[2], and Fabiano Dalpiaz[3]

[1] KU Leuven, Leuven, Belgium
estefania.serralasensio@kuleuven.be
[2] Università Politecnica delle Marche, Ancona, Italy
{p.sernani,a.f.dragoni}@univpm.it
[3] Utrecht University, Utrecht, The Netherlands
f.dalpiaz@uu.nl

Abstract. When many requirements co-exist for a given system, prioritization is essential to determine which ones have higher priority. While the basic prioritization algorithms result in a total or partial order of the requirements, it is often the case that the priority of the requirements depends on the context at hand. This is especially true in ambient intelligence systems such as smart homes, which operate in an inherently dynamic environment that may affect the priority of the requirements at runtime. For example, depending on the health status of a smart home inhabitant, safety may become more important than comfort or cost-saving. In this paper, we make three contributions: (i) we introduce a novel method for the contextual prioritization of requirements, (ii) we propose an online platform for prioritizing the requirements for a smart home based on our method, and (iii) we report on results from an initial evaluation of the platform and the prioritization method.

1 Introduction

Requirements prioritization helps to identify which requirements in a given set are the most important for a system and its stakeholders [4]. Prioritization is typically conducted during the design or evolution of a system to distinguish between critical and optional requirements.

The basic prioritization algorithms return a total or partial order of the requirements in the set; this occurs, for instance, with popular techniques such as the MoSCoW method (which distinguishes between must have, should have, could have, won't have) [7] or the Analytic Hierarchy Process (AHP) [17].

For some systems, however, the priority of the requirements changes at runtime depending on the context at hand. This is true for context-aware [3] and self-adaptive systems [14], which adjust their behavior to the ever changing environment wherein they operate. A major trigger for such adaptation is that changes in the environment affect the relative importance of the non-functional requirements (NFRs) [5], or qualities, of the system.

© Springer International Publishing AG 2017
A. Braun et al. (Eds.): AmI 2017, LNCS 10217, pp. 94–109, 2017.
DOI: 10.1007/978-3-319-56997-0_7

Ambient Intelligence systems and smart homes are a prominent example of systems that necessitate dynamic priority of NFRs and that can adapt their behavior based on the varying priorities. For example, if the health of a smart home inhabitant worsens, *safety* may become more important than *comfort* or *cost-saving*, and the home's behavior may switch to one where all monitoring devices are operational and the home becomes more intrusive by explicitly asking the inhabitant to provide information about her condition.

Existing work [8] proposes an approach for adaptive smart homes that relies on user- and context-specific priorities over NFRs. There, the smart home behavior is driven by an adaptive task model, which customizes the plans that the home carries out depending on the context and on user preferences. However, such approach provides no specific technique for eliciting those priorities.

In this paper, we address such limitations by proposing a novel elicitation technique for contextual priorities over NFR—that builds on and extends AHP—and by applying it to the smart homes domain. Specifically, we make three concrete contributions beyond the state-of-the-art:

– A method for the contextual prioritization of non-functional requirements that is intended for use by layman people with no expertise in prioritization.
– An online platform that supports the prioritization method for the context of smart homes. One key novelty of the platform is that it acts as a virtual proxy for the interaction between the analyst and the users.
– A preliminary evaluation of our platform with 25 users who employed the platform and judged how well the obtained adaptive smart home behavior complies with their preferences.

The rest of the paper is organized as follows. Section 2 discusses related work. The following sections describe our contributions: the contextual prioritization method (Sect. 3), the online platform (Sect. 4), and the evaluation results (Sect. 5). We conclude the paper and outline future work in Sect. 6.

2 Related Works

We review two strands of research that are relevant to the objectives of this paper: requirements prioritization and requirements elicitation in smart homes.

Requirements prioritization is defined the selection of the "right" requirements out of a given superset of candidate requirements so that all the different preferences of the end-users are fulfilled and the overall value of the system is maximized [16]. The purpose of any requirements prioritization technique is to assign values to distinct requirements that allow establishment of a relative order between them. To reduce costs, it is important to find the optimal set of requirements early, and then to develop the system according to this set.

There is a number of software requirements prioritization techniques [4], all of them with pros and cons. The 100-dollar test [13] requires to distribute 100 imaginary units (called dollars) among the individual requirements from the set: the more the dollars, the more important the requirement. Numerical assignment or grouping [11] requires to assign different labels to individual requirements that determine their priority groups (e.g., must have, should have, could have, won't have as in the MoSCoW method [7]). Ranking [4] requires the analyst to produce an ordinal scale of the requirements without ties in rank. The top-ten requirements approach [12] is useful when the wishes of multiple stakeholders are to be considered: each of them is required to list the ten requirements having the highest priority, and the results are then merged.

In our case, we have selected Analytic Hierarchy Process (AHP) [17], a systematic method that compares all possible pairs of hierarchically classified requirements in order to determine which has higher priority. The result is a weighted list on a ratio scale. AHP is one of the most complex methods but also provides fine granularity in the results and according to a recent survey [1], it is the most widely used technique. Moreover, it fits well our needs as our set of requirements is small (at most four requirements, as explained in Sect. 3).

As explained in Berander's survey [4], priorities should be determined by taking multiple aspects into account, including the importance of having the requirement (e.g., urgency or value), the penalty for not fulfilling the requirement [19], implementation cost, time, risk, and volatility. In this paper, we focus on the importance for the user, for we are interested in user-specific priorities.

Although many studies have been performed to study requirements prioritization in software engineering, the large majority of them only consider functional requirements since the prioritization process of NFR is harder [15]. Yet, NFRs are essential in AmI environments, for AmI systems are required to be sensitive to the needs of their inhabitants, anticipating their needs and behavior [18], and a viable way to do so is to use NFRs to guide adaptive behavior [8]. As far as our knowledge goes, no other work has proposed methods for prioritizing NFRs for smart homes or has considered contextual factors to adjust priorities.

In order to collect system requirements from end-users, many different techniques exist [21], such us interviews, task or domain analysis, focus groups, etc. Most of these methods require face-to-face communication, which has many benefits such as the ability to capture nuances in user requirements, but also several drawbacks. These techniques are time consuming, stakeholders are often incapable of expressing what they actually need (the say-do problem), and the interactions with software engineers may limit the exchange of information due to the influence of the engineer on the end user.

Other more advanced techniques can be used such as observation, monitoring or prototyping using living lab environments. An example is the Smart House Living Lab, which is fully equipped with the usual services of a conventional house where sensors and actuators are distributed in the living lab to offer a wide range of services [6]. However, living labs must be very flexible to offer all possible alternatives, and the development of environmental prototypes confronts many challenges such as cost-intensive and time-consuming experiments [2].

Finally, other alternatives have been proposed specifically to avoid these issues. For instance, Allameh et al. [2] propose the use of virtual environments to adjust the building design of a smart home according to users' preferences.

Their main outcomes are the possibility to use their approach for clustering target groups, the identification of living patterns, and the detection of spacial patterns. Unlike ours, their approach focuses mostly on functional aspects and does not explicitly consider the prioritization of NFRs.

3 A Method for Contextual Requirements Prioritization

We present our contextual prioritization technique, which can be used in the context of personalized systems that are able to customize their behavior to the preferences of the different users and contexts. A high-level illustration of the method is presented in Fig. 1 using the BPMN 2.0 notation.

The main goal of the technique is to obtain a contextual prioritization of non-functional requirements (NFRs); in other words, the priority of a NFR is not absolute but it depends on the context under consideration. Prior to system use, each user is expected to repeat the prioritization steps in order for the system to adapt to the individual preferences.

Three actors are involved: the *Designer* who prepares the environment for the prioritization activity, the *User* who expresses her preferences, and the *Platform* that algorithmically automates part of the process. The steps of our technique are described in the following and they are illustrated in Sect. 4.

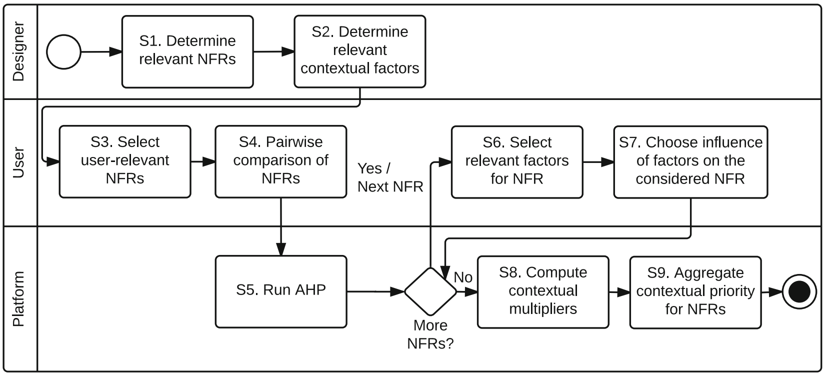


Fig. 1. Overview of the proposed contextual prioritization technique.

S1. Determine relevant NFRs. The designer of the system to-be determines the non-functional requirements (NFRs) to prioritize. We suggest to limit the number of NFRs to (at most) 4 to keep the process manageable, i.e., to minimize the required effort by the user.

S2. Determine relevant contextual factors. The designer identifies the contextual factors that may affect the NFRs' priority. For each factor, two descriptions are needed that denote when the factor holds and does not hold, respectively (e.g., "when it is hot weather" vs. "when it is cold weather").

S3. Select user-relevant NFRs. The user selects a sub-set of the NFRs to express which are the NFRs that she cares about. Not selecting a NFR corresponds to saying "I do not care at all about that NFR".

S4. Pairwise comparison of NFRs. The first prioritization activity employs classic AHP [17] and requires the user to perform a pairwise comparison of the NFRs to indicate their relative importance. To simplify the process, we use a simplified scale with 5 options to compare two NFRs: much less, little less, equally, little more, much more.

S5. Run AHP. Behind the scenes, the inputs of the pairwise comparison feed the AHP algorithms that returns the non-contextual relative importance of the NFRs. Together, the NFR importances sum up to 100%. We suggest to use the transitive calibration of the AHP verbal scale [9] to build the AHP matrix and compute a non-contextual relative importance of each NFR. A geometric progression is employed to calculate the priorities of the NFRs: the elements a_{ij} of the AHP matrix are equal to 1.0 when NFR_i and NFR_j have the same importance, 1.25 when the NFR_i is little more important than NFR_j and 2.441 when NFR_i is much more important than NFR_j.

S6. Select relevant factors for NFR. This step and the following one are repeated for each NFR that the user has not excluded. First, the user selects which factors affect the importance of the NFR. For simplicity, we limit the number of selectable factors to two.

S7. Choose influence of factors on the considered NFR. Multiple options exist depending on how many factors were chosen:

a. No factors are chosen: the NFR does not have contextual priority.
b. Only one factor is chosen for the NFR. The user has to assess the influence of the factor using the following scale: only important when the factor holds, more important when the factor holds, same importance regardless of the factor holding or not, more important when the factor does not hold, only important when the factor does not hold.
c. Two factors are chosen for the NFR. For the first factor (F_1), step S7b is executed. Depending on the answer for (F_1), step S7b is repeated for the second factor F_2:
   - "important only when F_1 holds": the user shall answer the question "when F_1 holds, how does F_2 affect the importance of the NFR?";
   - "important only when F_1 does not hold": the user shall answer the question "when F_1 does not hold, how does F_2 affect the importance of the NFR?";
   - "more/less important only when F_1 holds": the user shall answer two questions i. "when F_1 holds, how does F_2 affect the importance of the NFR?", and ii. "when F_1 does not hold, how does F_2 affect the importance of the NFR?";
   - "same importance regardless of the factor holding or not", step S7b is executed on F_2.

S8. Compute contextual multipliers. The platform automatically computes the effect of the contexts on the NFRs. For each NFR, the contextual multipliers are determined as follows:

- If "only if F" is selected, the multiplier holds corresponds to 1 when F holds and to 0 when F does not hold; formally, $M_F = 1.0$ and $M_{\neg F} = 0.0$;
- If "more if F" is chosen, $M_F = 0.\overline{6}$ and $M_{\neg F} = 0.\overline{3}$;
- For the "more if $\neg F$" option, $M_F = 0.\overline{3}$ and $M_{\neg F} = 0.\overline{6}$;
- For the "only if $\neg F$" option, $M_F = 0.0$ and $M_{\neg F} = 1.0$;
- If "equally" is chosen, the factor F is discarded: the user has actually stated that the factor has no contextual effect on the considered NFR.

When two factors are selected the contextual multipliers are the product of the multipliers of the individual factors for each combination of the factors holding or not. The syntax $M_{F_2|F_1}$ denotes the multiplier for F_2 which is selected in the context where F_1 holds (this is the answer to the questions of type "when F_1 holds..." in S7c); it corresponds to 0 when $M_{F_1} = 0$:

$$\begin{aligned} M_{F_1 \wedge F_2} &= M_{F_1} \cdot M_{F_2|F_1} \\ M_{F_1 \wedge \neg F_2} &= M_{F_1} \cdot M_{\neg F_2|F_1} \\ M_{\neg F_1 \wedge F_2} &= M_{\neg F_1} \cdot M_{F_2|\neg F_1} \\ M_{\neg F_1 \wedge \neg F_2} &= M_{\neg F_1} \cdot M_{\neg F_2|\neg F_1} \end{aligned}$$

S9. Aggregate contextual priority for NFRs. For each NFR, the contextual multipliers are applied to the non-contextual priority of the NFR x (P_x) from the AHP comparison as follows. First the platform finds the context y (boolean combinations of F_1 and F_2) having the highest multiplier M_{max}. Then, for each context y, the following equation results in the contextual priority value $CP_{x,y}$ for the NFR x in the context y:

$$P_x : M_{max} = CP_{x,y} : M_y \quad (1)$$

4 A Platform for Collecting Contextual User Preferences over Smart Home NFRs

Our aim is to apply the prioritization technique from Sect. 3 to regulate the behavior of a smart home according to its users' preferences. To do so, we developed a web platform that is used to conduct and to validate our method. Besides enacting the nine steps of our method (see Sect. 4.1), the platform collects metrics concerning how well the adaptive behavior of the smart home—guided by the contextual preferences—meets the users' expectations (see Sect. 4.2). Furthermore, the platform measures the users' perceived usability as well as information about users' demographics, education, and technical background.

4.1 Enacting the Contextual Prioritization Method

We created an online platform[1] that is structured as a questionnaire. The home page presents to the user the purpose of the questionnaire, i.e., "collecting users'

[1] https://goo.gl/ir65zM.

preferences concerning the behavior of their future smart home". The choice of using a website is made to maximize the ease of use and is enabled by the automated nature of the method described in Sect. 3. Should the target audience include people with little experience with computing, a human analyst can guide the users through the platform.

When the user begins the questionnaire, she is asked to select one to three aspects which she considers relevant. Such aspects are NFRs that are used to tune the smart home's behavior. We acted as designers (**S1** of our method), and chose three NFRs for the user to choose among, to avoid overwhelming her with too many questions:

– *Comfort*, representing the users' willingness to conduct tasks with minimal effort, and live in a comfortable environment (e.g., "I want my house to be always at the right temperature");
– *Efficiency*, representing the users' willingness to get things done quickly (e.g., "I want to skip breakfast if I have less time in the morning, and I want to take the fastest transport to go to work");
– *Utilities bill saving*, representing the users' willingness to pay lower utility bills (e.g., "I prefer to minimize the use of heating and air conditioning").

Table 1. The contextual factors supported by our platform.

Factor	Form 1 (factor holds)	Form 2 (factor does not hold)
Urgent tasks	I am in a rush	I do not have urgent tasks
Time period	I am not busy, e.g. I am on vacation	I am busy, e.g. I am working
Wealth	Money is an issue	Money is not an issue
Weather	It is good weather	It is bad weather

We also identified four contextual factors that are relevant for smart homes (see Table 1) as per **S2**: urgent tasks, time period, wealth, weather. For each of them, we defined two opposite forms that distinguish whether the factor holds (form 1) or not (form 2). It goes without saying that more factors could be considered, and our selection should be seen as illustrative.

If the user selects more than one NFR from the list (**S3**), she is asked to carry out a pairwise comparison of the selected NFRs, to perform the AHP [17]. For each couple of the selected NFRs (**S4**), the user has to state whether *(a)* the two NFRs are equally important, *(b)* one NFR is little more important than the other, *(c)* one NFR is much more important than the other (Fig. 2). The output is then processed by the platform that determines the AHP priorities (**S5**). In the example in Fig. 2, the non-contextual priorities are as follows: efficiency = 0.43, comfort = 0.35, utilities bill saving = 0.22.

Then, the user is asked to answer the question "Which of these aspects may affect how important <NFR-name> is for you?" to determine the influence of

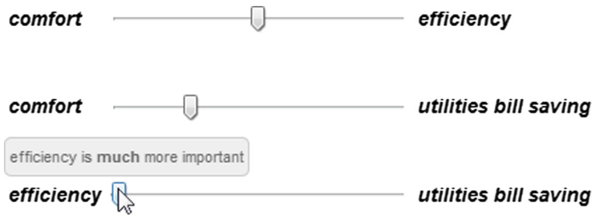


Fig. 2. The user pairwise compares the NFRs as part of the AHP process.

the context over the user's preferences. The user can select up to two factors (**S6**) for each NFR from the list shown in Table 1.

For each selected factor, the user is asked to state when it influences the NFR (see **S7** for the details): only when the factor holds (form 1 in Table 1), only when the factor does not hold (form 2), mostly when the factor holds/does not hold, or if the NFR has the same importance in both cases. Figure 3 shows an example related to the NFR "Comfort", when the user selected "Urgent tasks" as a contextual factor (**S7b**).

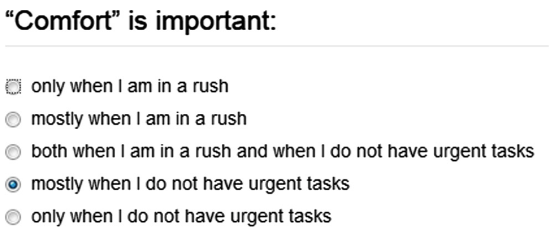
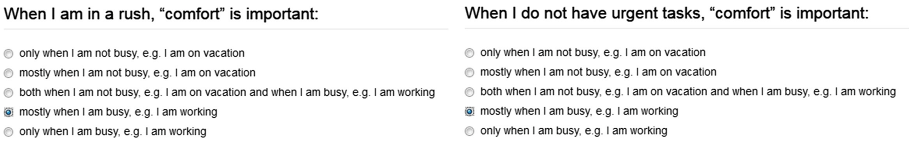


Fig. 3. Scoring how the factor "Urgent tasks" affects the NFR "Comfort".

Figure 4 shows an example assuming that a user selected both "Urgent tasks" and "Time period" as factors affecting "Comfort" (**S7c**). The user states that "Comfort" is important "mostly when I do not have urgent tasks". Consequently, the user is asked to rate how "Time period" affects "Comfort" when she is in a rush (Fig. 4a) as well as when she does not have urgent tasks (Fig. 4b). If the user had stated that "Comfort" is important "only when I do not have urgent task", the question of Fig. 4a would be omitted.

These choices determine the contextual multipliers (**S8**) for the various context. In Figs. 3 and 4, let F_1 be "urgent tasks" and F_2 be "time period". Figure 3 means "more if $\neg F_1$", while Fig. 4 indicates "more if $\neg F_2$" both when F_1 holds and when it does not hold. This leads to the following priorities:

- For context $F_1 \wedge F_2$, the multiplier is $0.\overline{3} \cdot 0.\overline{3} = 0.\overline{1}$
- For context $F_1 \wedge \neg F_2$, the multiplier is $0.\overline{3} \cdot 0.\overline{6} = 0.\overline{2}$
- For context $\neg F_1 \wedge F_2$, the multiplier is $0.\overline{6} \cdot 0.\overline{3} = 0.\overline{2}$
- For context $\neg F_1 \wedge \neg F_2$, the multiplier is $0.\overline{6} \cdot 0.\overline{6} = 0.\overline{4}$



(a) (b)

Fig. 4. Scoring how the factor "Time period" affects "Comfort", when the users selects that "Comfort" is important "mostly when I do not have urgent tasks".

The aggregated contextual priority is eventually defined by conducting step **S9**. In our example, the equation presented in Eq. 3 would lead to the following contextual priorities for the NFR "Comfort" (whose AHP priority is 0.35, as explained earlier in this section): 0.35 for context $\neg F_1 \wedge \neg F_2$, 0.17 for contexts $F_1 \wedge \neg F_2$ and $\neg F_1 \wedge F_2$, and 0.087 for context $F_1 \wedge F_2$.

4.2 Validating the Effect of the Priorities over Smart Home Behavior

The platform includes features in order for us to validate the obtained priorities by showing their effect on the behavior of a smart home, i.e., by activating different actuators. We designed three scenarios, one for each possible couple of NFRs. The smart home reacts to such scenarios according to the contextual priority assigned to the NFR, by using the framework presented in [8]. The scenarios are the following:

1. The home can wake the user up by gently opening the window blinds (comfort) or by activating the buzzer sound alarm (efficiency);
2. The home can refresh warm rooms by activating the air conditioning (comfort) or by opening the windows (utilities bill saving);
3. The home can activate the water heater (efficiency, as hot water is available more quickly) or employ the solar panel (utilities bill saving).

For example, in a context where comfort has higher priority than efficiency, the home will wake the user up by opening the window blinds instead of activating the buzzer sound alarm. For each scenario (thus, for each possible couple of NFRs), the platform shows two different contexts to the user: that where the first NFR has the maximum priority, and that where the second NFR has the maximum priority. The platform presents to the user the behaviors of the smart home in both cases, and the user is asked to express her agreement with such behavior on a Likert scale from 1 (strongly disagree) to 7 (strongly agree).

Besides the scenarios, the platform obtains further information on:

1. *perceived efficacy* of the platform for the user to express her preferences, via a 7-items Likert scale about agreement with the statement "The scenarios reflected the behavior I'd like for my smart home".

2. *usability* of the platform with the Usability Metric for User Experience (UMUX) [10] whose wording was customized for the platform as follows: *(a)* "The website enables me to express my preferences." *(b)* "Using this website is a frustrating experience." *(c)* "This website is easy to use." *(d)* "I have to spend too much time correcting things with this website."
3. *technical background* of the user. First, the user has to express her familiarity with Internet technologies (e.g., news websites and social networks), computer applications (e.g., word processors), and with programming languages via a Likert scale from 1 (not at all comfortable) to 5 (very comfortable). Then, the platform asks the user if she is working (or has worked) in the ICT sector. Finally, the platform asks the user if she is familiar with the "smart home" concept, by letting her select one of the following options: *(a)* "I never heard of smart homes" *(b)* "I heard the term, but I don't know what they are" *(c)* "I know what smart homes are" *(d)* "I am able to understand well how smart homes work (i.e. the technologies used)" *(e)* "I would be able to design/develop part of a smart home"
4. *demographics*, i.e., age, sex, country, and educational level.

5 Test Results

We report on a set of user tests concerning the perceived efficacy of our prioritization technique for the use case of a smart home, and the usability of the web platform that we developed. As described in Sect. 4, the web platform presents a questionnaire to the user; we discuss here the collected results concerning the obtained contextual priorities, the agreement with the proposed smart home scenarios, and the usability test.

Participants. The tests involved 25 users: 16 males and 9 females. The average age of the user is 31.96, with a standard deviation of 8.32 years. The users are from Belgium (11), Italy (6), and Spain (8). 11 users have a Ph.D. degree, 12 a master degree, 1 user has a bachelor degree and 1 user has a secondary school educational level. 20 users stated to have working experience in the ICT field: in particular, 9 of them stated to be able to develop components of smart homes.

Threats to Validity. The user tests that we performed to validate the prioritization should be considered preliminary due to the many threats to validity:

Conclusion. The small number of users does not allow to draw any statistically significant conclusion and has low statistical power. Also, due to the fact the participants are not native English speakers, the reliability of our measures (the questionnaires) may be limited. Although we tried our best to simplify the wording so to avoid misinterpretations, the threat is not nullified. Moreover, our choice to minimize the number of NFRs and factors to avoid overwhelming the users with too many questions (see internal validity) may affect the judgment of the users on the adequacy of the smart home behavior and on the ability of the platform to let them express their preferences.

Internal. The relevant internal threats categories for our tests are single group and social, according to Wohlin et al. [20]. Maturation threats are inevitable: while using the platform, some users may have kept motivated, while other may have been overwhelmed by the number of questions they were posed. To limit the effect of this threat, we limited the maximum number of selectable NFRs, contextual factors, and we employed simple questionnaires such as UMUX (that measures usability via four simple questions).

Construct. Design threats affect our tests: the link between NFRs and the adaptive behavior exhibited by the smart home was decided by the authors acting as designers. We did our best to choose scenarios that are clear illustrations of the prevalence of our NFR over another, but it is quite possible that the user's perception does not fully correspond to ours. Mono-operation bias also applies, given that we tested our contextual prioritization method only on specific behaviors of one smart home. To cope with hypothesis guessing, the home page of the website makes the context of our research clear.

External. The interaction of selection and treatment threat holds: we chose our subjects based on convenience sampling, and the obtained sample is certainly not representative for the whole population. While the sample group cannot be considered representative of current potential inhabitants of a smart home, the general increase in ICT skills of the human population makes the group more representative for the smart homes of the future. We need to repeat the tests with a larger and more representative audience to obtain more general results.

5.1 User Preferences and Validation of Scenarios

Concerning the distribution of the NFRs: 16 users selected comfort, 14 efficiency, and 17 utilities bill saving when asked to select the aspects relevant to them. In particular, 8 out of the 9 users who claim to able to develop smart home components selected comfort: according to smart home experts the comfort of the inhabitants is a key for a smart home to satisfy. On the contrary, most of the users who have just heard the term "smart home" (8 out of 11) selected Utilities bill saving: non-expert users seem to care more about the energy efficiency and the potential savings of living in an automated smart environment. This seems in line with having marginal knowledge on the fact that ambient assisted living goes beyond current trends in energy efficiency. Most users (14/25) have selected two NFRs, 7 have chosen only one NFR, and 4 have chosen all the three NFRs.

Table 2 highlights the selection of contextual factors per NFR. As expected, most of the users think that the factor wealth affects the importance of the NFR utilities bill saving: 14 out of the 17 users (82.4%) who selected such NFR. Urgent tasks is the most selected contextual factor for the NFR efficiency (78.6%): this is sensible, as a user would realistically prefer to do things quickly when she is in a rush. The most chosen factor for NFR comfort is time period: according to the users, the importance of comfort is mostly related to being in a working time period or on holiday (62.5%), although urgent tasks and wealth were also selected often (43.8%). Interestingly, weather was barely considered as an influencing

Table 2. Number (#) and percentage (%) of users selecting a factor per NFR.

	Comfort (n = 16)		Efficiency (n = 14)		Utilities bill saving (n = 17)	
	#	%	#	%	#	%
Urgent tasks	7	43.8	11	78.6	5	29.4
Time period	10	62.5	8	57.1	4	23.5
Wealth	7	43.8	4	28.6	14	82.4
Weather	2	12.5	0	0.0	2	12.5

factor, probably due to the fact that the users think of a smart home as a closed environment; we had included weather as a factor because the smart home described in [8] suggests a transportation means for reaching work.

Table 3 presents the average scores (on a Likert scale from 1 = strong disagreement to 7 = strong agreement) given by the users to the scenarios presented by the web platform. As explained in Sect. 4, the scenarios are computed by the platform to test whether the contextual priorities over NFRs lead to a behavior of the smart home that the users agree with. In each of the three scenarios, the home executes an action depending on the NFR with the highest priority. The results are weakly positive, with some scenarios being highly agreed upon and others obtaining neutral agreement ratings:

– The users give a score between weakly agreement and agreement (5.5) to opening the windows to refresh warm rooms (which happens when utilities bill saving is more important than comfort), while they are mostly neutral (4.22) on the converse scenario when comfort is more important and the smart home activates air conditioning;
– The users agree (6.11) with the scenario where the home employs solar panels to get hot water for the shower (this happens when the priority of utilities bill saving is higher than that of efficiency), while they are between neutral and weakly agreeing (4.57) on using the water heather when efficiency has higher priority;
– The users agree (6.22) that the home should wake them up by opening the blinds when comfort is preferred over efficiency; conversely, using the buzzer sound alarm in the same scenario is rated between weak disagreement and neutrality (3.9).

Note that most of the users selected utilities bill saving as an important factor, and we can notice higher agreement with scenarios where green and energy efficient actions are executed: opening the windows and using the solar panels.

Besides assessing the scenarios in isolation (as per Table 3), the users were asked to express their agreement with the following statement about the overall behavior of the smart home: *the scenarios reflected the behavior I'd like for my smart home*. The average score given by the users is 4.52 (standard deviation 1.45). This is inconsistent with the generally positive score assigned to the

Table 3. Average agreement of the users with the scenarios presented to them ($\overline{x}$), and standard deviation (σ).

Scenario		$\overline{x}$	σ
The home can refresh warm rooms by	Activating the air conditioning	4.22	1.62
	Opening the windows	5.50	1.71
To have hot water the home	Activates the water heater	4.57	1.59
	Employs the solar panel	6.11	1.45
The home wakes you up by	Activating the buzzer sound alarm	3.90	2.12
	Gently opening the window blinds	6.22	1.03

individual scenarios, and especially evident for the 5 users who state to know well smart home: they give an average score of 4 (neutrality, with a standard deviation of 0.89) to the statement, but they agreed with the three scenarios, with scores equal to 6 (std. dev. 0.82), 6.75 (std. dev. 0.43), and 5.6 (std. dev. 2.33). Our interpretation for such inconsistency is that, due to their expertise in developing smart homes, they may have expected the smart home to execute actions which are not covered by the current website.

5.2 Usability

Table 4 shows the average usability score for the platform using the Usability Metric for User Experience (UMUX) [10] framework. In a scale from 0 (lowest sense of usability) to 100 (highest sense of usability), our platform gets an average score of 66.50 (standard deviation 17.77). This indicates that usability is not particularly good, although we can notice quite some differences when analyzing sub-groups of the population based on their familiarity with smart homes. We discuss the results per group, although the findings should be taken with care due to the small sample size.

The 9 expert users (who state to be able to develop smart home components) are the most negative towards the web platform: the average UMUX score is 56.02 (standard deviation 17.58). 5 of such users agreed with the statement "Using this website is a frustrating experience", even if 3 of them agreed with the statement "This website is easy to use". A possible interpretation—that should

Table 4. The average UMUX score of the web platform ($\overline{x}$), with the standard deviation (σ).

Users	$\overline{x}$	σ
All (25)	66.50	17.77
Just heard the term "smart home" (11)	67.80	15.09
Know well smart homes (5)	82.50	8.08
Able to develop smart home components (9)	56.02	17.58

be confirmed with follow-up interviews—is that those experts are frustrated because the platform does not implement all the NFRs and smart home behaviors that they might expect. We chose to limit to three the number of NFRs and scenarios to avoid overwhelming users with too many choices, especially those who are not experienced with the field.

The 5 users who state to know well smart homes are the most positive towards the web platform: the average UMUX score is 82.5 and the standard deviation is 8.08 (the lowest). In fact, these are the users who agreed the most with the proposed scenarios and behaviors of the smart home.

The 11 users who have just heard the term “smart home” have an average UMUX score of 67.8 (standard deviation 15.09), quite similar to the total population.

6 Conclusions and Future Work

In this paper, we proposed an elicitation technique to compute contextual priorities over non-functional requirements (NFRs) according to end-users’ preferences. The technique combines the need of personalized systems to respond to end-users’ preferences with the goal of context-aware systems to adapt to the current context. In Ambient Intelligence, smart environments require both personalizations based on users’ preferences and context-awareness. We applied our prioritization method to such domain, using three NFRs (*Comfort*, *Efficiency*, and *Utilities bill saving*) and four contextual factors (*Urgent tasks*, *Time period*, *Wealth*, and *Weather*) to regulate the behaviors of a smart home.

We developed a web platform, structured as a questionnaire, to carry out user tests with 25 participants. First, users were asked to indicate their preferences over the proposed NFRs and contextual factors. Afterwards, users were asked to validate the obtained prioritization: three scenarios (refreshing the home, heating water, and waking the user up) were shown presenting alternative behaviors of a smart home depending on the obtained priorities of the NFRs.

The results are encouraging for the use of our elicitation technique in the context of smart homes: in general, the users agreed with the proposed scenarios, based on the context they selected through the contextual factors, and on their preferences expressed by filtering the NFRs. Of course, the number of participants does not allow to draw any statistically significant conclusion, and large-scale replications are necessary to obtain more solid results as well as to assess the generality of the prioritization technique beyond smart homes.

The tests also show clear room for improvements: in fact, while the users agreed with the individual scenarios, they rated close to neutrally the statement “the scenarios reflected the behavior I’d like for my smart home”. A possible explanation for this inconsistency is that the users—especially the more experienced ones with smart homes—would have expected some behaviors which are not included in the current implementation of the web platform.

An inherent trade-off exists for designers who aim to employ an automated platform for the collection of contextual requirements: that between ease-of-use and accuracy. Designers have to decide which and how many NFRs to let the user choose between, which contextual factors, and what scenarios. The challenge is to keep the prioritization simple enough that users do not feel overwhelmed, especially those with no experience in prioritization. Moreover, the platform could be improved by explicitly presenting the scenarios to the user before asking her to fill the questionnaire, so that she knows in advance the capabilities of the smart home and understands that the task at hand is to express preferences so that the smart home makes the "right" choice among existing alternatives.

At the end of the questionnaire, we also included an assessment of the users' perceived usability of the web platform. The average result (66.5) on the UMUX score (0 = lowest sense of usability, 100 = highest sense of usability) indicates a clear need for improvement. The average score is even lower (56.02) if we consider only the group of users who are smart home experts and able to use programming languages. Conversely, the users who stated they "know well smart homes" are rather positive about the user experience (82.5). While we chose a simple metric such as UMUX (four Likert-type questions) to avoid drop outs due to excessive complexity of the task, more in-depth qualitative studies are needed to assess what are the exact obstacles to usability.

We envisage that future work will engage two different research fields: Ambient Intelligence and Requirements Engineering. While the former community provides domain experience and can greatly benefit from contextual prioritization techniques that enable more dynamic and user-centric smart homes, further research in Requirements Engineering is necessary to build reliable algorithms. This paper paves the way for this interdisciplinary research collaboration.

References

1. Achimugu, P., Selamat, A., Ibrahim, R., Mahrin, M.N.: A systematic literature review of software requirements prioritization research. Inf. Soft. Technol. **56**(6), 568–585 (2014)
2. Allameh, E., Heidari Jozam, M., Vries, B., de Timmermans, H., Masoud, M.: Smart homes from vision to reality: eliciting users' preferences of smart homes by a virtual experimental method. In: The First International Conference on Civil and Building Engineering Informatics, pp. 297–305 (2013)
3. Baldauf, M., Dustdar, S., Rosenberg, F.: A survey on context-aware systems. Int. J. Ad Hoc Ubiquit. Comput. **2**(4), 263–277 (2007)
4. Berander, P., Andrews, A.: Requirements prioritization. In: Aurum, A., Wohlin, C. (eds.) Engineering and Managing Software Requirements, pp. 69–94. Springer, Heidelberg (2005). doi:10.1007/3-540-28244-0_4
5. Chung, L., Nixon, B.A., Yu, E., Mylopoulos, J.: Non-functional Requirements in Software Engineering, 1st edn. Springer, Heidelberg (1999)
6. Colomer, J.B.M., Salvi, D., Cabrera-Umpierrez, M.F., Arredondo, M.T., Abril, P., Jimenez-Mixco, V., García-Betances, R., Fioravanti, A., Pastorino, M., Cancela, J., Medrano, A.: Experience in evaluating AAL solutions in living labs. Sensors **14**(4), 7277–73111 (2014)
7. DSDM Consortium: DSDM public version 4.2 (2007). http://www.dsdm.org/version4/2/public

8. Dalpiaz, F., Serral, E., Valderas, P., Giorgini, P., Pelechano, V.: A NFR-based framework for user-centered adaptation. In: Atzeni, P., Cheung, D., Ram, S. (eds.) ER 2012. LNCS, vol. 7532, pp. 439–448. Springer, Heidelberg (2012). doi:10.1007/978-3-642-34002-4_34
9. Finan, J., Hurley, W.: Transitive calibration of the AHP verbal scale. Eur. J. Oper. Res. **112**(2), 367–372 (1999)
10. Finstad, K.: The usability metric for user experience. Interact. Comput. **22**(5), 323–327 (2010)
11. IEEE Computer Society: IEEE Recommended Practice for Software Requirements Specifications. IEEE Std 830–1993 (1994)
12. Lauesen, S.: Software Requirements: Styles and Techniques. Pearson Education, Upper Saddle River (2002)
13. Leffingwell, D., Widrig, D.: Managing Software Requirements: A Unified Approach. Addison-Wesley Professional, Boston (2000)
14. de Lemos, R., et al.: Software engineering for self-adaptive systems: a second research roadmap. In: de Lemos, R., Giese, H., Müller, H.A., Shaw, M. (eds.) Software Engineering for Self-Adaptive Systems II: International Seminar, Dagstuhl Castle, Germany, October 24–29, 2010 Revised Selected and Invited Papers. LNCS, vol. 7475, pp. 1–32. Springer, Heidelberg (2013). doi:10.1007/978-3-642-35813-5_1
15. Pergher, M., Rossi, B.: Requirements prioritization in software engineering: a systematic mapping study. In: 3rd International Workshop on Empirical Requirements Engineering (EmpiRE), pp. 40–44 (2013)
16. Ruhe, G., Eberlein, A., Pfahl, D.: Quantitative WinWin: a new method for decision support in requirements negotiation. In: Proceedings of the 14th International Conference on Software Engineering and Knowledge Engineering, SEKE 2002, pp. 159–166. ACM (2002)
17. Saaty, R.: The analytic hierarchy process–what it is and how it is used. Math. Model. **9**(3), 161–176 (1987)
18. Sadri, F.: Ambient intelligence: a survey. ACM Comput. Surv. **43**(4), 36:1–36:66 (2011)
19. Wiegers, K.E., Beatty, J.: Software Requirements, 3rd edn. Pearson Education, Upper Saddle River (2013)
20. Wohlin, C., Runeson, P., Höst, M., Ohlsson, M.C., Regnell, B., Wesslén, A.: Experimentation in Software Engineering. Springer, Heidelberg (2012)
21. Zowghi, D., Coulin, C.: Requirements elicitation: a survey of techniques, approaches, and tools. In: Aurum, A., Wohlin, C. (eds.) Engineering and Managing Software Requirements, pp. 19–46. Springer, Heidelberg (2005). doi:10.1007/3-540-28244-0_2