From Requirements Elicitation to Variability Analysis using Repertory Grid: A Cognitive Approach

Sangeeta Dey* and Seok-Won Lee†
*Department of Computer Engineering,†Department of Software Convergence Technology
Ajou University, Suwon City, Republic of Korea 443 – 749
*priya@ajou.ac.kr,†leesw@ajou.ac.kr

***Abstract*—The growing complexity and dynamics of the execution environment have been major motivation for designing self-adaptive systems. Although significant work can be found in the field of formalizing or modeling the requirements of adaptive system, not enough attention has been paid towards the requirements elicitation techniques for the same. It is still an open challenge to elicit the users' requirements in the light of various contexts and introduce the required flexibility in the system's behavior at an early phase of requirements engineering. We explore the idea of using a cognitive technique, repertory grid, to acquire the knowledge of various stakeholders along multiple dimensions of problem space and design space. We aim at discovering the scope of variations in the features of the system by capturing the intentional and technical variability in the problem space and design space respectively. A stepwise methodology for finding the right set of features in the changing context has also been provided in this work. We evaluate the proposed idea by a preliminary case study using smart home system domain.**

***Index Terms*—requirements elicitation, variability analysis, self-adaptive systems, repertory grid.**

I. Introduction

Increasing complexity in the execution environment has raised the interest in designing self-adaptive systems (SAS) whose behavior is contingent upon varying situation [1]. With the rapid advancement of technology, such systems are now well integrated with every aspect of human life, including user's comfort, leisure activity and other personal goals. The system is expected to understand the context of the user, the intention of the user in that context and behave accordingly. Thus the major driving force for designing such systems should be the user: the human. Different types of people come with different cognitive abilities. It is the requirements engineer's responsibility to support the users to express their needs in such a way that eventually leads to the design of a solution for the users. This challenge motivates us to shift our focus towards more effective personal requirements processes.

Requirements are believed to bridge the gap between the user's need (problem space) and the system design (design space). How a solution is designed, is always dependent upon how well the problem space has been captured and explored. Thus it is imperative to discover the scope of variations in the behavior of SAS and focus the design activities in the right direction at an early phase of requirements engineering (RE). A goal based approach for variability acquisitions has already been studied in [2] [3] [4] wherein the variability is discovered from the stakeholder's goals. However, such approach is based on a strong assumption that each stakeholder's cognitive ability would be able to support the intended complexity of the whole process of variability acquisition. In reality, for many complex domains it may be difficult for the stakeholders to describe their needs in a systematic way. The goal of this research is to provide a natural way to elicit requirements from the stakeholders and help designers to explore the intentional variability in the light of unintentional variability.

We propose a novel approach of using a cognitive technique for knowledge acquisition, Repertory Grid (RG) [5] for discovering the scope of variations in the behavior of SAS at an early stage of RE. Although RG has been proved to be an effective technique for eliciting stakeholders' mental models [6] [7], our study suggests that the effectiveness of this technique is not just limited to capture knowledge. With the help of the elicited knowledge it is even possible to extract the set of features of the system that are most appropriate for a given context. While the early phase of RE is mainly focused on the users/customers and their needs, the late RE activity is more concerned with exploring the possible design alternatives by using expert's or designer's knowledge. We discuss how RG can be useful for both the phases to capture the intentional variability and technical variability by building a problem space and design space respectively. These two spaces have been visualized as multi-dimensional in our work based on how various aspects influence the features in these spaces. We finally provide a stepwise methodology for discovering variabilities in the features of the system-to-be by capturing and analyzing various stakeholders' mental models.

The rest of the paper is organized as follows. In Section II, we discuss the existing work done to support the variability analysis and highlight our motivation for this work. The principle of RG is described in Section III. In Section IV and V, we discuss how problem space and design space are conceived and explored using RG. To provide better guideline for the requirements engineer and the designer, a stepwise methodology is proposed in Section VI. In order to evaluate the effectiveness of our approach we use a preliminary case study of smart home domain in Section VII. Finally, we conclude the paper with some remarks on our proposed idea and possible future work for improvement.

II. Background and Motivation

In the past decade, many researchers focused on how to satisfy a number of possible behaviors in order to meet the users' requirements. The notable work done so far can be categorized based on their different dimensions as follows:

A. Requirements Driven Adaptation in SAS

Introducing variability in the behavior of the system based on the changeable requirements is key challenge for any adaptive system. As argued by I. Jureta et al. in [9], the requirements problem in case of SAS is more complex compared to other domains as it involves exploration of a large problem space. Few efforts have been made to facilitate the RE of SAS and introduce the flexibility in the system behavior. Four different levels of RE have been identified by Berry et al. in [10]. The researchers argue that in order to make a system truly dynamic, some RE activity needs to be done at run-time by the system. Silva Souza et al. attempt to operationalize adaptivity by proposing a run-time monitoring framework which exploits the feedback loop control based on the satisfaction/failure or evolution of other requirements [11] [12]. The modeling and formalization of requirements of such systems have also been studied recently. Baresi et al. introduce the concept of fuzzy goal in order to model the variations of the system behavior by leaving some space for relaxation of non-crisp goals [13]. Based on the fuzzy logic, John Whittle proposes a new requirements specification language especially for SAS [14]. The intuition behind this approach is also capturing the scope of variations or relaxation of the system behavior in a formal way. However, most of this work is overloaded with complex formalization which hinders the natural way of the elicitation of user's need.

B. Feature Models Variability

The idea of using features for analyzing the commonality and variability originates from the concept of software product line. Feature Oriented Domain Analysis (FODA) [15] was the first attempt to model features for capturing and analyze the commonality and variability among applications in a domain. In terms of the origins of variability, many product line engineering approaches consider the domain experts as their main sources [16]. However, context can also play an important role to model a set of features. A feature can be considered as mandatory or optional based on the context at run time. The study by Hartman et al. shows the relation between the features and the context to support software supply chain [19]. It has been argued that, in order to accommodate a lot of information related to feature and context the semantics have been overlooked in case of feature modeling. Kannan Mohan and B. Ramesh focused on acquisition and management of knowledge related to variability in product family [18]. However, these approaches are extremely solution oriented. The variation points and the corresponding variabilities have not been considered from the requirements perspective.

C. Goal Based Variability

Significant work has been done on variability acquisition based on either problem descriptions [17] or annotated goal modeling [2] [3] [21] [23]. R. Ali et al. mainly focus on contextual goal model for those systems operating in changing context [20]. Alexei et al. extended i* modeling technique to accommodate all possible effects of domain variability by augmenting the goal model with context [22]. Liaskos et al. provided a list of variability concerns that are likely to be considered in most of the cases [2]. According to this work, the analyst is expected to consider relevant variability concerns before decomposing a high level goal into lower levels. It works well with domains where eliciting the concerns are straightforward. Example of such domain can be a patient assisting system [2] where a notification needs to be sent to the nearest nurse when a patient with hypotension tries to stand up on her own. In such case, if a guideline is provided, the variability concerns can be analyzed efficiently as the user can easily define what kind of notification she prefers or what conditions of the patient (*patient is almost up* or *patient is trying to get up*) should trigger this notification.

However, there are other complex domains like socio-technical systems, where eliciting these concerns is not straightforward. Let us consider smart home domain for instance, where the high level goal is to 'provide comfort in economic way'. For a simple feature like *Temperature Management* the user needs to think about various aspects while describing her intention. The user may want the temperature of her room to be managed automatically at a comfort level whenever she is at home. However, she might want to sacrifice this comfort when the price of electricity goes up during peak hour. Furthermore, she does not care about high price of electricity when she has special guests at home. So, altogether the analysis of such concerns in the light of various contexts or aspects becomes extremely complex. In order to deal with this challenge, it would be wise to take help of a cognitive technique to elicit such requirements instead of totally relying on the users and the analyst's cognitive ability.

III. Repertory Grid Technique

RG is an interviewing technique which was devised by George Kelly based on his Personal Construct Psychology theory [5]. According to this theory humans create representations about events or objects in their minds by using contrasting poles or *constructs*. For example: a person who needs to decide on holiday destination can think of multiple alternatives. Each of these alternatives is an *element* of the grid and is rated based on contrasting poles like *Expensive vs Cheap* or *Too far vs Close* etc. on a scale of '1 to 5' or '1 to 7' as required.

This technique has already been used in knowledge engineering by many researchers [6] [7] [8]. However, its usage can be extended even further. We can design RG in such a way that it can capture the stakeholders' intentions or expectations of the system-to-be in various contexts. The advantage of such technique is that, it provides a good visualization of the mental representation of a subject. This kind of easy

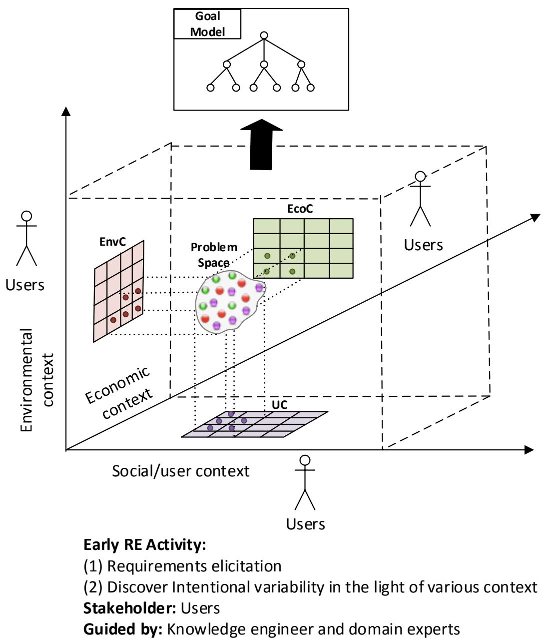


Figure 1: Problem Space Conceptualization

way to capture and explore a subject's mental model can be useful for our research problem. If we can capture the problem space using RG, then it can eventually be effective to discover the variability in the subject's need. Furthermore, to accommodate the variability discovered in the problem space, a solution space needs to be explored. RG can be proven to be useful to capture various design options that the designers are thinking about in order to meet the user's requirements and analyze the applicability of those options. Finally, by using our methodology it becomes easier to deal with the challenges mentioned in Section I such as:

1) Intentional variability of the users can be analyzed by capturing their mental representations of requirements.

2) In order to support the user's requirements, all possible design variabilities can be explored by capturing the design space with the help of expert's knowledge.

3) Moreover, each of the RGs can be saved throughout the life-cycle of the system. This can enhance the traceability by containing all the requirements and design related knowledge (rationales of decisions) in it.

IV. Managing Problem Space

In this section, we explain our idea of capturing problem space using an example of applying RG to a socio-technical system. We choose such a domain because of its complex nature of pertaining social, technical and economic aspects to reach efficiency. As a result, performing a variability analysis for such system is more challenging. We illustrate how to overcome some of the difficulties by using RG.

A. Problem Space Conceptualization

Problem space of a system-to-be is a visual representation of the concerned problem. In order to design a system, the first step is to capture the user's mental representation of her needs. For a socio-technical system, these requirements mostly vary based on various contexts. Three main dimensions along which a problem space can be conceived are- social, environmental, economic contexts as shown in Fig.1. For instance, in case of smart home whose main goal is to 'provide comfort to the user in the most economic way', the following three aspects play essential role in determining the requirements and exploring the problem space.

1. User's activity/social context *(UC)*: whether the user is at home or not, he is working or sleeping at home etc.

2. Environmental context *(EnvC)*: whether the day is sunny, rainy or gloomy etc.

3. Economic context *(EcoC)*: The price of electricity, user's range of expenditure for electricity etc.

When and how a feature is required depend on all the above mentioned contexts. We use RG to elicit the requirements and the impact of these contexts. The whole process has to be guided by the knowledge engineer and the domain experts. The combinations of all the intentions of user in the light of various contexts automatically generate the intentional variability in the problem space. Based on the acquired knowledge of the user and her preference, a goal model is usually constructed in the next phase. However, we mainly focus on the first step of elicitation of requirements rather than modeling them following a formal technique.

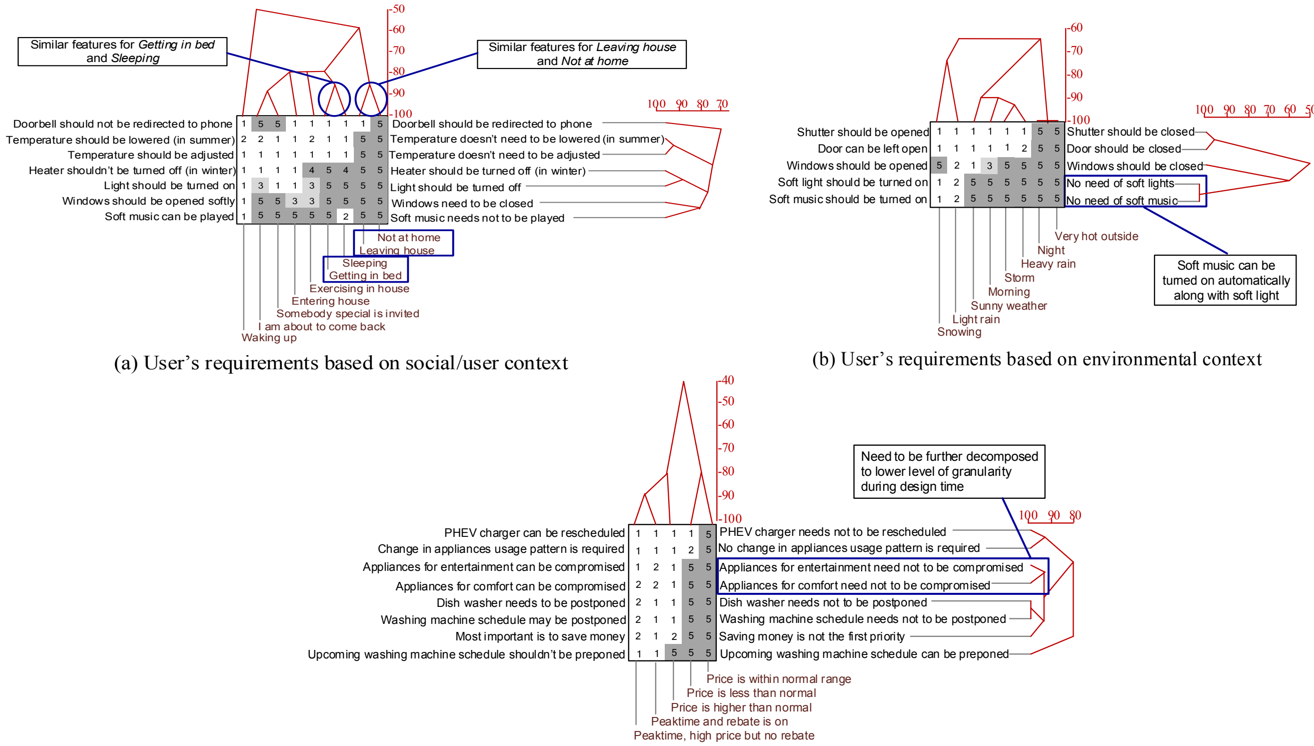
B. Construction of Problem Space using RG

RG needs to be designed intuitively in such a way that the user feels comfortable to relate with real-world situation and express her requirements. There can be two ways of eliciting variability concerns related to the requirements:

(i) Focus on each feature and then identify the concerns that are related to that feature [2]. Therefore, for each feature the user needs to consider all the contexts that may impact the concerned feature. This might be challenging for a user with a general cognitive ability.

(ii) The user can be asked to focus on the context or situation first and then try to relate features she needs in that situation. This motivates us to design RG where the various contexts are considered as *elements* and then the subjects differentiate those contexts based on the features that are needed in those situations. To form such RGs we conducted an experimental session with few of our colleagues who are not expert in smart home domain. We chose such subjects to understand the usefulness of our approach in case of people with less or no experience at all in the problem domain. We made three grids for each subject as shown in Fig.2(a)-(c). The RGs shown in Fig.2 are from the same subject.

1) Eliciting requirements based on social/user context (UC): To understand the emergent property of a system as a whole, we use full RG where we allow the subjects to identify various social/user contexts. Then we let them



(a) User's requirements based on social/user context

(b) User's requirements based on environmental context

(c) User's requirements based on economic context

Figure 2: Problem space captured and explored using RG .

think of a number of features that they might need in those contexts. As shown in Fig.2(a), each element is rated against many contrasting poles. For instance, the element *Leaving house* is rated '5' (on a scale of '5') against the contrasting poles *Temperature should be adjusted* (left pole) and *Temperature does not need to be adjusted* (right pole). This implies that, in the context *Leaving house*, temperature does not need to be managed any more. Similarly, for any context, the corresponding column clearly shows what features are relevant to it and what all are unnecessary. Interestingly, if we slightly change our perspective to analyze the grid, we can see that each row depicts the context (variability concern) that are relevant to the corresponding feature. For example: *window needs to be opened softly* when the user is *waking up* and *needs to be closed* for rest of the contexts except while *exercising in house* and *entering house*, as the subject has rated '3' for the same to express that he does not need any automatic behavior of window specifically for these two contexts. Furthermore, the clusters (made by RG) at the top and right side may also be useful, though analyzing the implications of such cluster highly depends on the ability of the knowledge engineer and the designer. In the provided example, *Sleeping* and *Getting in bed* are almost 90% similar in terms of the features needed in those two contexts. This might motivate a designer to introduce a *Sleep* mode with similar features while designing the feature model. Another valuable finding is, for obvious reason this mental representation of the preference would vary from subject to subject. This can be extremely helpful for designing personalized adaptive system.

2) Eliciting requirements based on environmental context (EnvC): Different users may have different personal preferences of experiencing comfort based on the environmental situation. We aim at eliciting such preferences naturally by using RG. The subjects first need to identify the environmental situations along which their requirements or preference can vary and then rate each of them against the related feature. For instance, based on the contexts like *Sunny weather* or *Heavy rain* the features *Open window* or *Close window* will be activated. Though most of these relations can easily be supported by common knowledge, we find it useful to let the subjects describe their requirements in natural language. This might leave a scope of discovering some quality attributes or totally *Unknown Unknown* concepts [27]. As shown in Fig.2(b) the subject mentions about *Soft light to be turned on* and *Soft music to be turned on* while snowing or raining outside. This puts one level of quality constraint on the general feature *Light* and *Music* as they need to be of soft type when it snows or rains outside. Moreover, these two features are identical in terms of their triggering condition as per the hierarchical cluster made by RG. This indicates that, the automation of these features should exhibit the same pattern.

3) Eliciting requirements based on economic context (EcoC): Another important and more complex dimension of the problem space is economic context. We do not go deep into

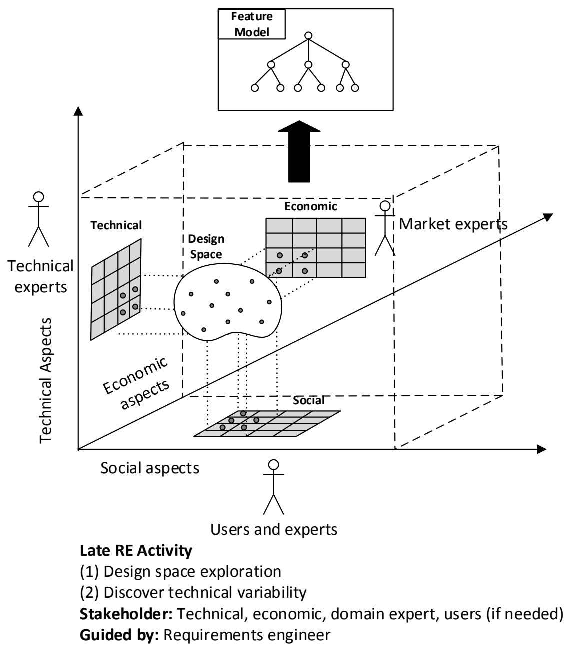


Figure 3: Design Space Conceptualization

the details of pricing and market features as they are part of Smart Grid family. However, we are interested in analyzing the impact of various economic context like when *Price is higher than normal*, *Price is within the normal range* etc. As shown in Fig.2(c), some of the contrasting poles are at different level of granularity. Such as: *Appliances for comfort can be compromised*, *Appliance for entertainment can be compromised* are quite abstract, whereas *Dish-washer can be postponed* or *Washing machine can be postponed* are very specific. This kind of conflicts further need to be clarified while constructing the design space. Similar to the previous grids, in this case also, for any given economic context the corresponding column lists up the user's desire in that particular context.

V. Managing Design Space

In this section, we illustrate how the design space of a socio-technical system can be conceived by capturing the expert knowledge using RG.

A. Design Space Conceptualization

The design space is a guideline for a designer to identify required decisions, the alternatives and their correlations in order to solve a given problem [25]. In [24] a design space of socio-technical system is conceived from RE perspective. However, once the problem space is explored, the requirements problem of SAS is nothing but a design and decision making problem [9]. This motivates us to solve the problem from design perspective in the later phase of RE. In order to identify the design options and their applicability, we need to consider not only the technical aspects but also the related social and economic aspects. This suggests visualizing the design space as a three-dimensional space as shown in Fig.3. However, selection of such dimensions may vary based on the concerned domain. For instance, in the case of a smart grid domain we can identify the social (user's needs), technical (technical opportunities) and economic aspects(market options, pricing techniques) very easily. Whereas domains like Ambulance Dispatch System or Flood Prediction System might not have much concern related to economics, instead current laws or political views may be influential for those systems.

B. Capturing Design Space along Multiple Dimensions

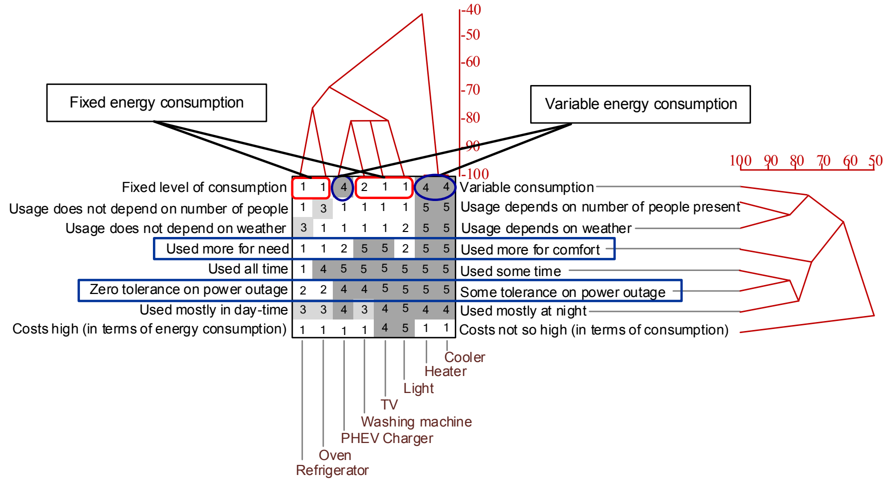
Based on the identified dimensions, the design space can be captured with the help of experts' knowledge using RG. The first step towards constructing the design space is to set the focus of interest based on the analysis of the problem space. The RGs already designed in Section IV, can be helpful in finding the key features that the designers should be interested in. We now describe how design variability can be explored to support the identified key features.

1) Social dimension of design space: Most of the social aspects of design space can easily be elicited using the process mentioned in Section IV. However, any granularity issue like the one identified in previous section needs to be solved before moving towards the exploration of technical variability. In order to find the scope of variations to meet the high level goal *provide comfort in the most economic way*, it is required to fully capture the users mental representation about how comfort or need is related to the appliances the user uses. During our study, we interview the subjects using RG about various electric appliances they use at home and the nature or pattern of their usage. The result of such interview from the concerned subject is shown in Fig.4(a). Some of the notable information regarding the usage of the appliances is:

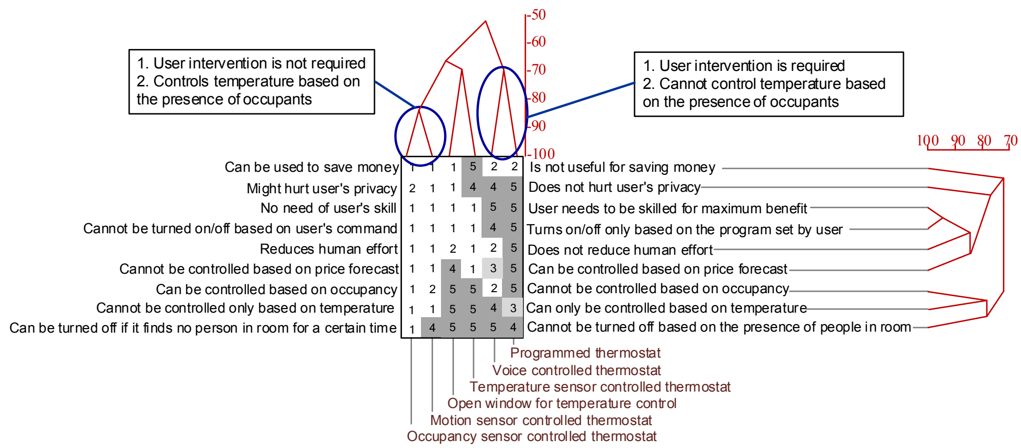
- Zero Tolerance on outage: Refrigerator, Oven
- Some tolerance on outage: Light, TV, Cooler etc.
- Used mostly for comfort: Washing machine, Heater, Cooler, TV.
- Used for need: Refrigerator, Oven, Light etc.

The appliances which are mostly used for comfort and have some tolerance on power outage can be turned off if the price of electricity is very high and user agrees to compromise with comfort in such a situation. This can be one among many ways to reach the high level goal of the system. Although it may be argued that such information can easily be extracted using common knowledge, it is preferred to avoid any bias of the designer. Also, discovering the social aspects from a user's perspective is always effective. This way it is possible to resolve the granularity issues and clearly identify the scope of possible variations.

2) Technical dimension of design space: One of the major sources of variable behavior of a system-to-be is the technical variability. For any key feature, there can be multiple ways to design it. A technical expert chooses one of those options during design time based on some facts or rationales best known to her. One major challenge faced during system design



(a) Social knowledge captured by RG



(b) Technical knowledge captured by RG

Figure 4: Design space captured and explored using RG

is the traceability of such design decision and their rationales. RG has already been proven to be very effective in order to reduce vaporization of such knowledge [26]. The key features identified in the problem space are: *Light Management*, *Temperature Management*, *Redirect Door bell to phone*, *Music* etc. (identified from the constructs of RGs in Section IV). Technical variability to support *Temperature Management* is explored in Fig.4(b). The subject of our study is a PhD scholar whose research interest is in RE of Smart home and Smart Grid domain. She first identifies six ways to manage temperature and then analyzes the commonality and variability of those options using RG. Each of the clusters at the top has its own implications which are summarized in Table I. Such analysis is extremely important to trace back to the rationales of design decision any time during the system's life cycle.

3) Economic dimension of design space: Various market options and their impact on reaching the system goal should be analyzed during design time. For instance, different kinds of electricity pricing technique may affect the electricity market and overall design of smart grid differently. An economic expert can explore such economic aspects using RG in the similar fashion as technical aspects. Since the focus of our case study is on smart home domain, we assume that a realtime pricing technique is already chosen as a part of smart grid design.

VI. Stepwise Methodology for Variability Analysis

After constructing problem space and design space, the task of exploring the scope of variations reduces to the selection of suitable features from the design space based on a given context and user's preference for that context. In Fig.5 we summarize the whole process of variability analysis by providing a stepwise methodology.

1) *Step1: Construct the problem space:* Construct the problem space focusing on mainly three kinds of contexts: User context *(UC)*, Environmental Context *(EnvC)* and Economic Context *(EcoC)* using RG as shown in Fig.2(a)-(c). The user should be guided by a knowledge engineer while constructing the problem space.

Table I: Analysis of Technical Variability

Percentage of similarity	Elements	Common Feature
85%	Occupancy sensor controlled thermostat, Motion sensor controlled thermostat	User intervention is not required, can be controlled based on the occupancy of the room, can be useful for saving money, might hurt user's privacy
70%	Temperature sensor controlled thermostat, Open window for temperature management	User intervention is not required, cannot be controlled based on the occupancy of the room, can only be controlled based on the temperature
70%	Programmed thermostat, Voice controlled thermostat	User intervention and skill is required, may be useful for saving money, do not hurt user's privacy

2) *Step2: Record and analyze the identified conflicts or contradictions:* As this interviewing technique allows the subjects to describe their needs in natural language, the analysis of RG may show some conflicts or contradictions in the mental representation of the subject. Such conflicts should be resolved by further discussion in this step. In our case study, we found a potential conflict while eliciting the environmental contexts and corresponding needs. The subject mentioned *Morning*, *Snowing*, *Light rain* as different contexts and rated them differently against various features (Fig.2(b)). However, her demand in case of a *Rainy morning* or *Snowy morning* was not mentioned. The subject further confirmed the higher priority of contexts *Snowing* or *Raining* compared to *Morning*, which means, in a context like *Raining in the morning*, her requirements in context *Raining* will override the requirements in context *Morning*.
3) *Step3: Identify the key features needed to be designed:* Based on the preferences captured in the problem space, designer needs to identify the key features of the system-to-be. This can easily be done by analyzing the constructs of each of the RGs. For instance, from Fig.2 the key features can be identified as: *Light Management*, *Temperature Management*, *Redirect door bell* etc.
4) *Step4: Construct the design space:* After identifying the key features, a designer needs to explore all possible ways to technically support the user's requirement. For example, as shown in Fig.4(b), temperature can be managed in multiple ways and all those options are explored using RG by capturing the technical expert's mental model. It is also important to explore the design space along all the related dimensions which are influential on the design decisions.
5) *Step5: Identify the user's intentions in a given situation:* For any given situation, the first activity is to decompose that situation in three different types of context. For instance, in a situation like 'due to extra workload, the user is working at office even after 9pm and the price of electricity is normal', the *UC* is *User not at home*, *EnvC* is *Night time* and *EcoC* is *Price is within the normal range*. The designer should focus on the respective columns of each of these contexts in RG to find the user's intentions.
6) *Step6: Identify the most feasible set of features for the given situation:* Once the required key features and the user's preferences for a given context are identified, the designer just needs to explore the design space in order to find the most feasible way to support those requirements. She can be guided by a requirements engineer during this activity as satisfaction of the qualitative requirements can also influence the selection of the best set of features. For instance, if there are multiple ways to manage the freshness of the air of a smart home, it is needed to find out the best among them by analyzing the preference or constraints mentioned by the user while constructing the problem space (explained with scenarios in Section VII).

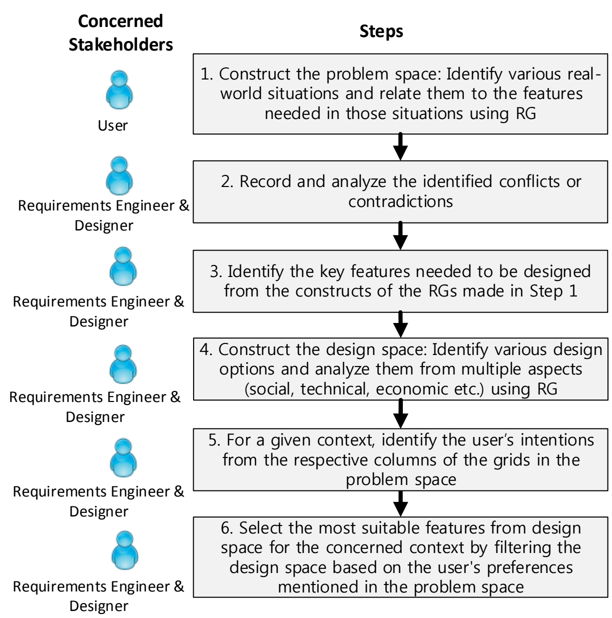


Figure 5: Stepwise methodology for variability analysis

VII. Evaluation

In order to formally evaluate *How and Why* this methodology will help us to reach the goal (as mentioned in Section I), we follow the validation process as proposed in [28] [29].

A. Study Questions

To prove the effectiveness of our proposed approach, the following two study questions need to be answered:

1) How and why the proposed methodology helps eliciting requirements for a complex SAS?

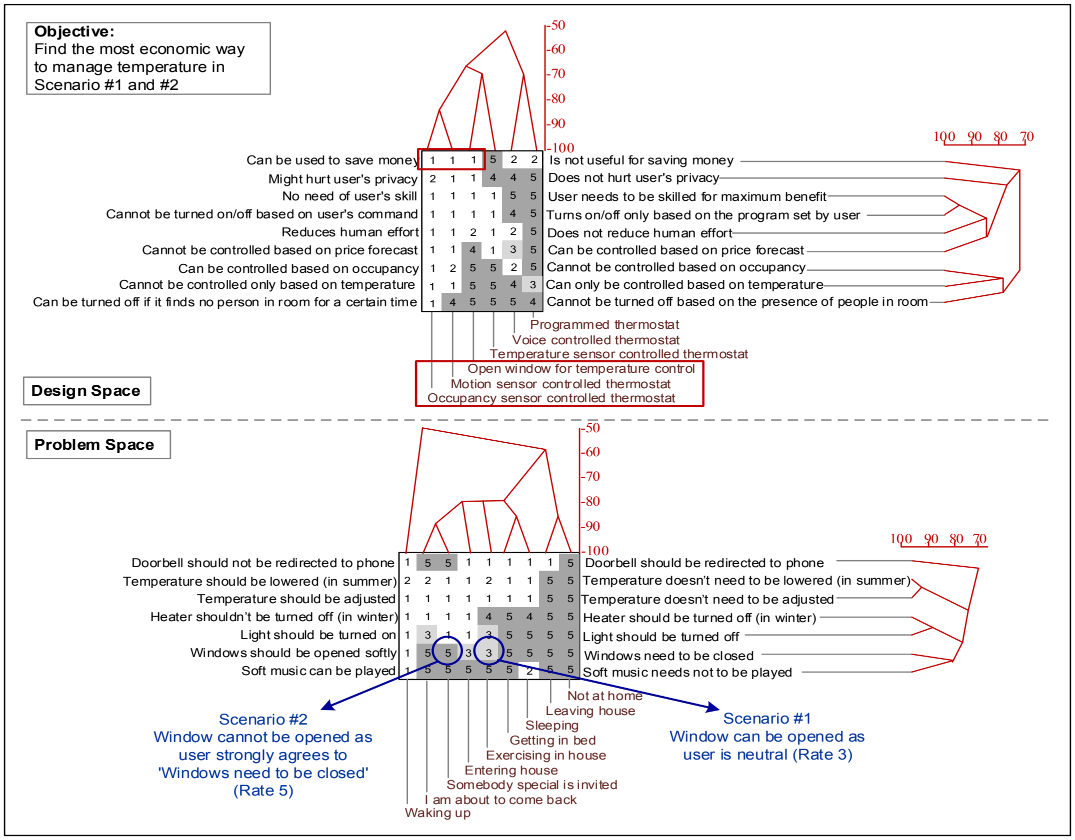


Figure 6: Result of variability analysis

2) How and why the proposed methodology provides better guideline to the designers than the traditional methods to form and explore the design space of such systems?

B. Case Study Propositions

Case study propositions are the assertions that need to be examined to answer the identified study questions. The general proposition (GP) to support the study questions is: "The proposed methodology can achieve its research goals because it provides precise guideline to capture and explore the intentional and technical variability in various context using a cognitive technique RG". More specific propositions (SP) are as follows:

SP.1.1 The proposed methodology helps the users to describe their requirements by focusing on distinct situations from social, environmental, economic perspectives and then relating them to features using RG.

SP.1.2 As the RGs made by the stakeholders are nothing but their own mental models, they can help us to detect the possible conflicts or contradictions which need to be resolved by further clarification at an early phase of RE.

SP.2.1 With the help of the problem space made by the users, it becomes easier for the designer to focus on the overall set of required features of the system.

SP.2.2 To support each of the key features, the designer can explore the possible design options (technical variability) and their commonality or variability using RG.

SP.2.3 Given the problem space and design space, it becomes easier to filter the most feasible design options for any given context based on the user's preference.

These propositions will be supported by the evidences after conducting the preliminary case study of smart home domain.

C. Units of Analysis

Unit of analysis is the actual source of information that measures the achievement of study proposition. The units of analysis that we use in this study are:

- Repertory grids capturing intentional variability
- Potential conflicts
- Set of required key features
- Repertory grids capturing design variability
- Filtered design alternatives

D. Smart Home Case Study Result Summary

To verify the effectiveness of the proposed approach, we apply our methodology to extract the variable features from the captured knowledge based on two scenarios.

Scenario #1: On a sunny morning, when the weather is pleasant outside, the user is exercising at home. But, due to

Table II: Evidence Collection through Requirements Elicitation and Variability Analysis Summary Sheet.

Activity	Captured Evidence	Supported Proposition
Step1: Construct problem space	**Three RG files captured intentional variability from the user by relating real-world situations and corresponding required features:** **Identified social contexts**: Not at home, Leaving house, Sleeping, Getting in bed, Exercising in house, Somebody special is invited etc. Related features: Management of light, temperature, soft music, door bell, window. **Identified environmental contexts**: Snowing, Morning, Sunny weather, Light rain etc. Related features: Door, Window, Shutter, Soft light, Soft music. **Identified economic contexts**: Price is in normal range, Price is less than normal, Price is higher than normal, Peaktime and rebate is on, Peaktime but no rebate. Related features: Compromise comfort and entertainment if needed, Reschedule dish washer and washing machine etc.	GP, SP.1.1
Step2: Identify and resolve conflicts	**Detected potential conflict in the problem space:** Temporal factors (morning or night) mixed up with environmental factors (snowing, raining). Resolved by prioritizing environmental factor than temporal one in case of morning and vice versa in case of night. Keep window closed if it rains heavily in the morning. Do not open window in case of light rain if it is night.	GP, SP.1.2
Step3: Identify the key features	**The overall set of required features that the designer should focus on, is identified by consolidating the constructs of all the RGs:** Temperature management, Light management, Music, Auto open/close window, doors, shutters, Reschedule washing machine, dish washer.	GP, SP.2.1
Step4: Construct design space	**Two RG files captured the scope of variations along the dimensions of design space:** **Social Variability:** Identified appliances for comfort: Washing machine, Heater, Cooler, TV etc. Identified appliances with some tolerance on outage: Light, TV, Cooler etc. These appliances can be turned on/off based on the price of electricity. **Technical variability:** Identified six ways to manage temperature and analyzed them as shown in Table I. **Economic variability:** Not explored as part of this case study.	GP, SP.2.2
Step5: Identify user's intentions in *Scenario #1* and *#2*	**The features to focus on for *Scenario #1* and *#2* are identified from the respective columns of RGs made by user:** Temperature needs to be adjusted. Light needs to be turned on. Window needs to be closed only for *Scenario #1*.	GP, SP.2.1, SP.2.3
Step6: Filter the design space based on user's preference	**Available options to manage temperature in economic ways are: open window, motion sensor controlled thermostat and occupancy controlled thermostat. The filtered alterntaives are:** Scenario #1: Manage temperature by opening window. Scenario #2: Manage temperature by thermostat and motion/occupancy sensor attached to it.	GP, SP.2.3

some reason the price of electricity is very high at that time. *Scenario #2:* This scenario is almost the same as the previous one except that now the user is not exercising at home as he is busy attending a guest at his house.

The intentional variability of smart home domain has already been explored by following *Step1-3* in Section IV. However, due to space constraint we only explored the technical variability regarding *Temperature management (Step4)*. Therefore, the main focus of this case study is to (perform *Step5-6*): 'Find the most economic way to manage temperature at home and provide comfort to the user in *Scenario #1* and *#2*'. The design decision was made based on the following facts:

1. User expects to save money when the price of electricity is high by sacrificing some of his comfort (Fig.2(c)).

2. Heater and cooler are used mostly for comfort (Fig.4.(a)).

3. However, for both the scenarios, the user wants the temperature to be managed. (Fig.2(a))

4. Among various ways to manage temperature inside smart home, the most economic ones are:

Opt.1: Occupancy sensor controlled thermostat.
Opt.2: Motion sensor controlled thermostat.
Opt.3: Open window to manage temperature.

5. We should also consider user's other quality requirements like privacy and check if he has already put any constraints on the applicability of such options. In Fig.2.(a) one constraint is found on the feature *Open window automatically*. The user strongly disagrees to open the window when *he has guests at home*, whereas he remains neutral about it in case *he is exercising at home*. This leaves the scope to use the feature *open window to manage temperature* in order to provide comfort in the most economic way in case of *Scenario #1*, whereas in case of *Scenario #2* a thermostat should be working based on a motion sensor or occupancy sensor attached to it. The core part of this analysis is depicted in Fig.6.

E. Evidence Collection

We map the outcomes of each step (in terms of the units of analysis) of the proposed methodology to support/reject certain propositions. Such evidences are captured in *Requirements Elicitation and Variability Analysis Summary Sheet* (Table II). In the case of our domain of interest, it is less effective to perform a quantitative analysis. Instead, it would be helpful to focus on how well the methodology supported all of the propositions in a systematic way.

F. Discussion

Based on the results of the case study, it is evident that the proposed methodology is capable of reaching its research goal and discovering scopes of variations in the system's behavior. This works well especially to elicit the quality requirements (e.g., privacy, security etc.) that put complex constraints on the design decisions. Two main challenges faced during the case study were *When to stop refining the RGs* and *How to manage multiple grids and their rich contents manually*. The right time to stop elicitation of requirements using RG can be decided based on the quality of the contents of the grids. If any conflict is found after analyzing the first set of input from the user, that needs to be clarified before stopping the elicitation. The second concern can be addressed by automating the process of variability analysis based on the captured knowledge.

However, the validity of this case study can be threatened by the ability of the knowledge engineer as the quality of variability analysis depends on how well the knowledge is captured. Moreover, as this is not a formal experiment, we have to rely on analytical generalization instead of statistical generalization. In case of any complex domain involving multi-dimensional aspects, this methodology can be applied to increase understandability of the problem space. The promising results open up new research questions as *how to elicit requirements from multiple stakeholders and reach a consensus for multi-dimensional problem domain*.

VIII. Conclusion

We propose a novel approach of using a cognitive technique RG to not only elicit requirements from the users but also analyze the design knowledge of the experts. We illustrate how RG can be used effectively to explore multi-dimensional problem space and design space of complex adaptive systems in a socio-technical environment. Finally, the proposed methodology provides a precise guideline to the requirements engineers and designers to perform a variability analysis on the captured knowledge. We envision RG as a technique to filter the design options based on the intentional variability and user's preferences in various contexts. Such variability analysis can be helpful especially for the green-field domains. In the near future, we plan to validate the proposed approach by an empirical study. A tool support can also be provided for aggregating inter-participant grids in case of multiple stakeholders.

Acknowledgement

This research was supported by Basic Science Research Program through the National Research Foundation of Korea (NRF) funded by the Ministry of Education, Science and Technology (No. 2013R1A1A2009801).

References

[1] De Lemos, Rogerio, et al., Software Engineering for self-adaptive systems: A second research roadmap. *Software Engineering for Self-Adaptive Systems II. Springer Berlin Heidelberg,* pp.1-32, 2013.

[2] Liaskos, Sotirios, et al., On goal-based variability acquisition and analysis. *Requirements Engineering. IEEE,* pp.79-88, 2006.

[3] Yu, Yijun, et al., Configuring features with stakeholder goals, *Proc. 2008 ACM Symp. on Applied computing. ACM,* pp.645-649, 2008.

[4] Liaskos, Sotirios, et al., Goal-based behavioral customization of information systems. *Advanced Information Systems Engineering. Springer Berlin Heidelberg* pp.77-92, 2011.

[5] Gaines, Brian R., and Mildred LG Shaw, Knowledge acquisition tools based on personal construct psychology. *The knowledge engineering review 8.01.* pp.49-52, 1993.

[6] Shaw, Mildred LG, and Brian R. Gaines, Requirements acquisition. *Software Engineering Journal 11.3* pp.149-165, 1996.

[7] Niu, Nan, and Steve Easterbrook, Discovering aspects in requirements with repertory grid. *Proc. International workshop on Early aspects at ICSE. ACM*, 2006.

[8] Niu, Nan, and Steve Easterbrook, So, you think you know others' goals? A repertory grid study. *Software, IEEE 24.2*, 2007.

[9] I. J. Jureta et al., The Requirements Problem for Adaptive Systems. *ACM Transactions on Management Information Systems (TMIS)*, 2014.

[10] Berry, Daniel M., Betty HC Cheng, and Jia Zhang, The four levels of requirements engineering for and in dynamic adaptive systems. *International Workshop on Requirements Engineering Foundation for Software Quality*, 2005.

[11] Silva Souza, Vtor E., et al., Awareness requirements for adaptive systems. *Proc. 6th International Symp. on Software engineering for adaptive and self-managing systems*, *ACM*, 2011.

[12] Souza, Vtor E. Silva, Alexei Lapouchnian, and John Mylopoulos, (Requirement) evolution requirements for adaptive systems. *Proc. 7th International Symp on Software Engineering for Adaptive and Self-Managing Systems*, *IEEE*, 2012.

[13] Baresi, Luciano, Liliana Pasquale, and Paola Spoletini, Fuzzy goals for requirements-driven adaptation. *Requirements Engineering*, *IEEE*, 2010.

[14] Whittle, Jon, et al., RELAX: a language to address uncertainty in self-adaptive systems requirement. *Requirements Engineering*, pp. 177-196, 2010.

[15] Kang, Kyo C., et al., Feature-oriented domain analysis (FODA) feasibility study. *(CMU/SEI-90-TR-21). Technical Report, SEI/CMU*, 1990.

[16] Frakes, William, Ruben Prieto, and Christopher Fox, DARE: Domain analysis and reuse environment, *Annals of Software Engineering*, pp.125-141, 1998.

[17] Salifu, Mohammed, et al., Using problem descriptions to represent variabilities for context-aware applications. *International Workshop on Variability Modeling of Software-intensive Systems (VaMoS)*, 2007.

[18] Mohan, Kannan, and Balasubramaniam Ramesh, Ontology-based support for variability management in product and families. *Proc. Hawaii International Conference on System Sciences. IEEE.* 2003.

[19] Hartmann, Herman, and Tim Trew, Using feature diagrams with context variability to model multiple product lines for software supply chains. *Software Product Line Conference*, *IEEE*, 2008.

[20] Ali, Raian, Fabiano Dalpiaz, and Paolo Giorgini, A goal-based framework for contextual requirements modeling and analysis. *Requirements Engineering*, 2010.

[21] Liaskos, Sotirios, et al., Exploring the Dimensions of Variability: a Requirements Engineering Perspective. *VaMoS 7. Springer Berlin Heidelberg.* 17-26, 2007.

[22] Lapouchnian, Alexei, and John Mylopoulos, Capturing Contextual Variability in i* Models. *iStar.* pp.96-101, 2011.

[23] Lapouchnian, Alexei, and John Mylopoulos, Modeling domain variability in requirements engineering with contexts. *Conceptual Modeling-ER. Springer Berlin Heidelberg.* pp.115-130, 2009.

[24] Chopra, Amit K., et al., Protos: Foundations for engineering innovative sociotechnical systems. *Requirements Engineering. IEEE.* pp.53-62, 2014.

[25] Brun, Yuriy, et al., A design space for self-adaptive systems. *Software Engineering for Self-adaptive Systems II. Springer Berlin Heidelberg.* pp.33-50, 2013.

[26] Tofan, Dan, Matthias Galster, and Paris Avgeriou, Reducing architectural knowledge vaporization by applying the repertory grid technique. *Software Architecture. Springer Berlin Heidelberg.* pp.244-251, 2011.

[27] Sutcliffe, A., and Sawyer, P., Requirements elicitation: towards the unknown unknowns. *Requirements Engineering. IEEE.* pp.92-104, 2013.

[28] Lee, Seok Won, and David C. Rine, Case Study Methodology Designed Research in Software Engineering Methodology Validation. *SEKE.* pp.117-122, 2004.

[29] Yin, Robert K., Case study research: Design and methods. *Sage Publications.* 2013.