Chapter 1

Introduction

Since last two decades, in the software engineering discipline, engineering requirements have been gaining momentum by considering many parameters. It is clear that a large portion of the software developments have failed because of uncertain requirement explanations (Konrad and Cheng, 2004; Palomares et al., 2014). Many models have been developed to capture the importance of requirements analysis (Bleistein et al., 2006). An analyst can, in general, start from the basic point of gathering requirement, interpreting the requirements as the requirement analysis has to be done in first phase of the development cycle. However, this is highly dependent on the experience of the analyst (Choppy and Hatebur, 2005; Harrison et al., 2015). The recurring problems in the requirements are usually addressed based on requirement patterns. Furthermore, the utilization of the requirement patterns in any system can begin with the recognizable proof of the patterns and its reusability of the scope inside the instantaneous nature of the systems, thereby creating the requirement specifications and allowing the analyst to reuse the requirements instead starting from scratch. For writing the software requirements, requirement patterns may provide situational based specific solutions. Today, most of the requirements in requirements engineering have recurring problems, which are not addressed in a systematic way. Although the usage of requirement patterns helps the analysts to a larger extent, there is a demand for developing reliable and consistent methodologies that deal with the recurring problems. Towards this, the domain pertaining to context aware systems is gaining importance among researchers and engineers.

In day-to-day applications, context based requirements can widely improve many industries, in particular, the software industry. Many times, analysts and experts continue to work on providing the solutions in an arbitrary way until the situation becomes out of control. A context aware system is in demand due to the reaction of individual segments and its circumstances (Chen and Kotz, 2000; Hsieh and Lin, 2014). The scope of these applications deals with their surroundings pertaining to system associated design varying from standard to pervasive, mobile computing (Ye et al., 2015). This necessitates for reusable structure in the requirements of ubiquitous computing which forms an option to work for context aware system and its requirements of reusability (Alegre et al., 2016). Since context aware computing and
its functions are still in the testing stage and often involve rework or modifications, there is a need of formalizing context properties in the system. There are ways to provide solutions to recurring problems. The proposed work formalizes the issues using requirement pattern template and methodological steps to provide suggestions and improvements.

1.1 REQUIREMENT PATTERN

A requirement pattern is a format and manual for composing a specific kind of requirement for example, execution, reinforcement and recuperation, report, question, etc. Every pattern indicates what data has to be gathered for the specified requirement, such as what components to include and what to prioritize, etc. The idea of a pattern is entirely based on some basics (Castro et al., 2002). In light of the kind of requisite, a single requirement is used to gather a patterns; every pattern has to deliver the most exact and low level information as expected to gather.

The objective of requirement pattern is to empower the expert to gather higher quality requirements quickly and easily with less effort. The better requirements result from the usage of patterns in view of the fact that requirement patterns point out the issues that reasonably closer to examination and incidentally block the experts from ignoring crucial focuses (Cheng and Atlee, 2007). In some cases, patterns give direction on the most proficient method to compose a necessity and can likewise outfit case requirements. Patterns help expert or analyst profitability by giving an incredible beginning stage. The expert starts with a significantly effectively composed pre-requisite, as opposed to starting without any preparation. Some particularly complex examples give orderly directions to assemble the required data (Hagge et al., 2005).

On the other way requirement pattern characterized by Fowler (1997) states: “A pattern is a thought that has been valuable in one functional connection and will be most likely helpful in others”. To address context based challenges, patterns need to be taken after the recorded data available in the pattern property. A pattern might take after recorded credits to comprehend and give important answer for definite software product (Withall, 2007).

Stephen Withall initially presented the thought of requirement examples in his book "Programming Requirement Patterns" (2007). As indicated by Withall, a high rate of the elements found in any business framework are regular to some other sort of framework and just a moderately little rate of the necessities separates every framework
for distinctive purposes and commercial enterprises. In contemplating requirement particulars for an assortment of tasks, Withall saw that up to half of the necessities composed were basic to all ventures. With this data, Withall understood that creating necessity patterns could spare venture groups and business examiners critical exemption enhancing the nature of the requirements.

1.1.1 TEMPLATE FOR REQUIREMENT PATTERN

When all is said and done, Gamma et al. created outline design pattern and then Dwyer et al (1999) used to plan a requirement design in light of the configuration design template. Requirement pattern template proposed and addressed for pervasive computing by Konrad et al (2002). Then pattern based elicitation created by Renault et al (2009). The reusability idea of patterns substantially preserved time in development and gives flexibility in organization through which the need of conventional development procedure is impressively moderated (Rashid et al., 2015; Chung et al., 2012). Additionally this dissertation has bought context attributes and its properties to propose a new template for requirement engineers to determine the issues and implement with various circumstances.

1.2 CONTEXT BASED SYSTEMS

Ubiquitous systems in general assume a noteworthy part in registering advances for human machine interaction (HMI), and in particular at whatever point the context is changing much of the time with applications such as pervasive and versatile computing gadgets (Debes et al., 2005). In day by day exercises, where a few circumstances and individuals, connected with substantial measure of cooperation and responsibility, can be considered as a decent occasion for context. Such context can be installed into electronic gadgets and empowered like versatile advertisement with remote communication advancements for example wearable sensors to detect pulse, heat (Bauer, 2003; Bleistein et al., 2006).

Context aware system mentions to a computing concept in which the conduct of individual segments is controlled by the self-discovered circumstances. The degree with which these applications communicate with their surroundings significantly surpasses the commonplace framework/environment pattern regular to most cutting edge figuring. The systems has an exceedingly effective outcome on a specific
application part either the last needs to adjust in light of changing outside conditions or it depends on assets whose accessibility is subjected to ceaseless change. Context aware system first come out with the front line in the mid 1990's with the presentation of cell phones. Want et al (1992) active badge utilized infra-red correspondence between identifications worn by clients and sensors set to screen clients' developments and forward phone calls to them. These applications perform exercises going from just showing data to the client about his present area by joining a file index to a space to be utilized as a whiteboard by clients in the room (Want et al., 1995). Finally this concludes that there is a need for requirement pattern in context based system which is given below.

- **Problem:** Context aware requirements contains ambiguous, incomplete and sometimes uncertain requirements in nature.
- **Solution:** Requirement pattern with context aware support
- **Benefits:** Less effort during elicitation, save time and improve quality.

### 1.3 CONTEXT AWARE PATTERNS

The ordinarily accommodated requirement elicitation procedures are very little powerful in light of the fact that it doesn't have all around defined format. This prompts decreasing the likelihood of project success. Requirement pattern were getting unmistakable quality among investigation gatherings to offer software researchers in recognizing, breaking some assistance with downing and sorting out necessities of a context based framework system (Palomares et al., 2014). Frameworks empowers to recognize patterns, use safeguarding of self-rules and core configurations of the data sources, joining in the logical criteria. For examples, the effects of context are displayed in wording of context types and repeating content within it (Barrachina et al., 2012).

Essential connections are encompassed by the importance of context data. These essential contexts are personality, time, movement i.e. activity and location (Schilit et al., 1994). Taking into account context properties, a frame work is administrations to the client (Davies and Venkatesh, 1995). Context aware patterns can be a capable way to streamline the requirement designing procedures because they catch demonstrated contextual information through their template. There are works identified with setting contextual framework and patterns considered in this section. Each one of these works manages requirement pattern otherwise known as design pattern in the context domain.
PABRE (Pattern Based Requirement Elicitation) investigates the utilization of necessary reusability as per pattern strategy to fortify initial phase of elicitation by software experts, who involve beside a customer (Renault et al., 2009). Konard et al (2004) made specification patterns for real time in terms of normally utilized temporal logic's. **Requirement Designs in Knowledge Fusion**, the patterns are utilized to mean the elicitation steps to the information sources from the structures to affirm the full frameworks helpfulness (Smirnov et al., 2015). AWARENESS is architype, which addresses three patterns namely: event, control and action. To be particular, in the occasion control-activity, the context sources and administrators' chain of command and the activities designs can be associated conveniently in the headway of setting context facilities (Costa et al., 2005). OpenEHR combines contradictory areas like pervasive computing, agents and inter-operability (Cardoso et al., 2013). CAPOMF (Context Aware and Pattern Oriented Machine Learning Framework) is a machine learning for context aware system with the pattern framework (Ravindran et al, 2014). All these works deal with patterns in the context domain.

1.4 PATTERN BENEFITS

The commercial ventures are moved to reuse worldview to increase different difficulties. In this proposition, underlying causes of these challenges have been attended to for the result of context aware pattern for systems.

1.4.1 CHALLENGES

There are important discussion with the current state of context based requirement patterns research. To bringing the effectiveness of the pattern-based approach to software engineering would be difficult to organize in a systematic way and identify suitable context requirements.

Requirements patterns based on contextual research has not been developed fully. There are significant obstacles in the availability of context based information to organize, context tools, support the retrieval, application and refinement of requirements patterns. Furthermore, as Naish and Zhao (2011) work on developing classification schemata for requirements patterns (RePa) shows, finding a generalized way to organize requirements patterns is difficult and highly subjective. There are powerful mechanisms available to support the context-sensitive retrieval of reusable
artefacts from large reuse libraries – with recommender systems research offering a particularly promising approach – but with a few exceptions (Analogical Reuse in particular). These mechanisms have yet to be developed into practical tools to support the reuse of requirements patterns in practice.

Adding up with the detection and abstraction of further requirement patterns, the RePa community will need to investigate the relationships between the available requirements patterns in order to find systematic ways of organising the captured requirements knowledge, as well as to identify tools which can support pattern-driven approaches to requirements engineering.

1.4.2 BENEFITS

However, pattern-based approaches to requirement-level reuse, have proven popular because: Pattern, as an approach to reuse, both the theory and practice of software analysis and software design.

<table>
<thead>
<tr>
<th>Assumed Benefits</th>
<th>Extent to which realized</th>
</tr>
</thead>
<tbody>
<tr>
<td>Development of analysis model</td>
<td>Context aware system development will be worked out with the easy way. Only context requirements processed with the help of context attributes.</td>
</tr>
<tr>
<td>Pattern template to emphasize context attributes</td>
<td>Context aware system has context attributes to be addressed in an extensive way. The template is realized to recognize the reusability.</td>
</tr>
<tr>
<td>Context requirements – pattern methodology for a context aware system</td>
<td>Different patterns which elaborate the methodology to realize the context aware requirements in an effective way.</td>
</tr>
<tr>
<td>Performance of the context patterns</td>
<td>Quality of the context patterns and its usage with real world evaluated with the different approaches.</td>
</tr>
<tr>
<td>IEEE Characteristics of non-functional requirement</td>
<td>Nonfunctional requirements are addressed to ensure the quality.</td>
</tr>
</tbody>
</table>

Table 1.1 Benefits of context patterns

The cost to abstract a context aware pattern is relatively low; patterns are fine-grained and abstract, and so can readily be mixed-and-matched with other patterns to
compose complex requirements artefacts. In principle, context patterns can be organized into pattern languages, which are well-structured libraries of related pattern and collectively influence the requirement expert through deal with new requirements artefacts in a pattern-driven manner. Table 1.1 provides the benefits and what extent can be realized in the context aware system.

1.5 RESEARCH MOTIVATION

Today, most of the requirements are having recurring problems which is addressed to a certain extent and not in complete workout. On the other hand, context aware systems and context based requirements are in day to day applications which enhance the software industry worldwide in each year. Analyst and experts continue to work on providing solution without a formalized way until the situation becomes out of control. There are ways to provide solution to recurring problems. Requirement patterns research is one answer to the question: “How can reuse be achieved at the requirements level? The question is motivated by the often-cited claim that “reuse can improve the quality of software engineering artefacts”, and by the work of Boehm (1991) which suggests those “errors that occur earlier in the software lifecycle are most costly errors to fix” (since they have consequences further downstream): the pay-off from reuse can, therefore, be increased significantly if reuse can be achieved to resolve or avoid these more costly errors. Reuse is thus considered as an important part of requirements engineering research.

The term context in software engineering is initially presented by Schilit et.al (1994) and characterized "as Information about individual's nearness to environment, for example, area and personalities are close-by individuals and items". They suggest to setting as area, distinguishes adjacent individuals and their protests and changes them. In general, context are classified in to three parts namely: processing, client and physical context (Schilit et al., 1994). Whereas Chen and Kotz (2000) included the further class of connection as temporal setting. As per the detected data and stages of customized intuitiveness, few more additional portions have been included as dynamic and detached context. Then dynamic setting independently modify the behavior of the systems. Detached setting introduces the connection of data to the client chooses the best approach to modify the conduct.
Another work planned physical and virtual context with virtual online gathering help as per expansion of Abowd et al. (1999) study and exploration of their work (Winogard, 2001). Van Laerhoven et al. (2001) projected client determined action setting in view of the quickening information. Kofod-Petersen and Cassens (2006) anticipated while learning concentrated way to deal with concentrates numerous qualities of client setting and created five classes of context as: natural setting, individual connection, social setting, errand connection and spatio-fleeting setting.

Requirements patterns were grabbing conspicuousness among investigation gatherings to offer research specialists in recognizing, breaking some assistance with downing and sorting out necessities of a context based framework system (Palomares et al, 2014). Frameworks are empowered to recognize repetitions. Patterns use shielding of independences with inside configurations of the context sources joining in the relevant measures, specific to context environment. In pattern the properties of context are appeared regarding repeating context data standing in it.

1.6 RESEARCH OBJECTIVES AND QUESTIONS

The purpose of this research effort is to report the issues in formalizing context aware system to support reusability, design a requirement pattern template for handling the context aware system requirements and analyze the performance of the proposed model.

**Research objective:** To investigate how context aware requirement patterns might be shaped and research their impact on performance assessment.

This forms the basis for two research questions (RSQ):

1. *Is it possible to formalize context aware system’s requirements using requirement pattern?* (RSQ-1)
2. *How to identify a methodology and checking the performance evaluation of the context aware requirement pattern?* (RSQ-2)

To response these (RSQ-1 and RSQ-2), the accompanying targets are considered:

**Objective 1:** To study, analyze the need of requirement pattern for context aware system.

**Objective 2:** To propose a new template named CaRePa - Context aware Requirement Pattern (pronounced Care-Pa).
Objective 3: To design a methodology for adopting context-aware requirement pattern which is a CaRePa methodology.

Objective 4: To propose a framework, which decompose the requirement into pattern that uses problem frames to support reusability as mentioned DRAP-PF. This framework is a detailed part of a portion in CaRePa methodology.

Objective 5: To evaluate the performance framework of CaRePa for different cases using fuzzy inference system, ontology and survey approaches.

In view of the consequences of the observational examination, the hypothetical lens is modified and an exploratory design is proposed. Besides, fuzzy, ontology methods depict that how successfully setting context aware patterns in requirements are assessed and displayed.

1.7 STATEMENT OF PROBLEM

It is obvious from the current studies that no former exploration and system has been produced with the end goal of assessing the CaRePa for contextual systems. It is significant to add a structure tending to both CaRePa template and CaRePa methodology. A large amount of study reports have motivated on pattern and contextual environment based on design oriented patterns. Here the problem addresses the requirement pattern in the contextual requirements and assesses utilizing the customary measurable procedures. Besides, it is intrigued to utilize learning and adoptable methods through a mix of different strategies like: fuzzy, ontology etc.

In this theory, it ought to be noticed that, to the best information of the specialist, there are no current studies that explored the CaRePa template and CaRePa methodology based on the source of ontology and fuzzy method. Subsequently, the point of this dissertation is to recommend a requirement pattern for measuring different context aware system, methodology and performance based on statistical, fuzzy-ontology approaches as far as enhancing the context properties. All through this proposal, the importance of CaRePa template and performance assessment have been concentrated for context requirements taking into account CaRePa exploration and context aware system.
1.8 EMPHASIS OF THESIS

The center of this dissertation work is to plan and build up a requirement pattern to context system for giving properties of context aware problems existing in the real world. The research has been carried out in the following ways as mentioned in Figure 1.1. The state of art mainly covers [i] context aware patterns and key findings [ii] modeling Context Aware Requirement Pattern (CaRePa) template [iii] CaRePa methodology. [iv] The role of DRAP-PF framework in CaRePa. [v] CaRePa performance evaluation. The design of CaRePa methodology in context aware system considered is a base framework for the research work and model with ontology and fuzzy to analyze the performance of patterns in modeling.

![Research Methodology Diagram]

Figure 1.1. Research Methodology

So providing a requirement pattern in designing context aware system to handle context requirement issues with the help of CaRePa. Patient Monitoring System considered as a case study to demonstrate the CaRePa methodology steps. The practical
issues are analyzed in context gathering with the help of context reasoning using SPSS Tool, IEEE830 verification with PABRE, Fuseki server, SPARQL and MATLAB.

1.9 CONTRIBUTIONS TO THE RESEARCH AND THESIS OUTLINE

The foremost involvements of this thesis work are illustrated as given below:

1. To study and analyze the need of requirement pattern for context aware system.

2. To propose a new framework named CaRePa - **Context aware Requirement Pattern** (pronounced Care-Pa) which has CaRePa template.

3. To design a methodology for adopting context-aware requirement pattern which is a CaRePa methodology.

4. To propose a framework, which bring the pattern using problem frame approach to support reusability.

5. To evaluate the performance framework of CaRePa for different cases using fuzzy inference system, ontology and survey approaches.

1.9.1 THESIS OUTLINE

This thesis is sorted out in seven parts and the structure of the theory and exploration stream is introduced in Figure 1.2.

Chapter 1 presents an overview of requirement pattern in software engineering, pattern template, context aware patterns, research inspiration, purposes and outline of the proposal. Part 2 literature gives point by point writing review in three sections; first one relevant to research on requirement patterns, second one relevant to research on pattern templates and the third one relevant to research on context aware system and its related patterns. Based on this literature review gaps are identified and the system is produced. The exploration structure exhibited in Chapter 3 investigates the CaRePa template and the importance of context properties. Chapter 4 explains the CaRePa methodology has seven context patterns and its logical connectivity. Chapter 5 deals with the DRAP-PF, which has problem frame approach for evaluating the problem domain of context requirements under pattern environment. The performance evaluation of CaRePa formalized by way of using fuzzy, ontology and survey approaches are elaborated in Chapter 6. At long last, the summary, future enhancement and consequence of the proposal are discussed in Chapter 7.
Figure 1.2. Thesis structure and Research Flow