REQUIREMENTS ELICITATION
USING FRAMENET FRAMES

Thesis submitted for the degree of
Doctor of Philosophy
at the University of Leicester

by
Mahwish Kundi

Department of Informatics
University of Leicester

April 2018
REQUIREMENTS ELICITATION USING FRAMENET FRAMES

Mahwish Kundi

Abstract

Natural language-based use cases remain the key source of requirements elicitation despite the common problems associated with specifications, namely incompleteness, inconsistency and ambiguity. Various solutions such as viewpoints, goals, questionnaire, group meeting, interviews, and use case, have been developed to reduce the communication gap among stakeholders. However, a significant gap exists between theoretical and practical aspects of requirement engineering. This gap could be reduced by semantic agreement among all the stakeholders. This thesis presents a novel approach in order to tackle the textual description problems. It exploits linguistics patterns FrameNet frames in the concept description process. The proposed approach is systematically evaluated using qualitative and quantitative techniques. The qualitative study consists of experimental study and interviews. These studies are carried out to evaluate the role and impact of FrameNet frames in use case elicitation. The quantitative study takes into account a broader range of expert users so as to investigate the impact of FrameNet, and to identify whether the users (technical and nontechnical) consider the information of FrameNet useful or equally useful.

The results of the study have proved that use of frames in use case-based elicitation has enhanced the quality of use cases and lead to a clearer understanding of the domain concepts covered in a use-case. The proposed solution helps to acquire comprehensive and thorough information by exploring and identifying the missing requirements. This also results in an initial evaluation of the proposition explaining, that using frames as a reference model in requirements specification, improves the completeness, consistency, and clarity of the resultant requirements.
Acknowledgement

In the name of Allah, the Beneficent, the Merciful.

I would like to thank my supervisors Dr. Neil Walkinshaw and Dr. Ruzanna Chitchyan for their supervision and guidance on improving my project. I am thankful to my co-supervisor Dr. Fer-Jan de Vries for his support and help throughout the period of PhD.

Special thanks to my family and friends in their support. I would like to thank my friend Dr. Noshin Gul, Dr. Muhammad Younas, Maryam Al Hinaiand and Mr. Muhammad Amir for their guidance in my thesis writing.

A profound thanks to my sponsor Abdul Wali Khan University, Mardan and Govt. of Pakistan for funding me for the whole duration of the course.
Dedication

I would like to dedicate this thesis to my Mother and late Father.
Publications arising from this thesis


Contents

1. Introduction ......................................................................................................................... 1
   1.1 Overview .......................................................................................................................... 1
   1.2 Problems and Motivation ............................................................................................... 2
   1.3 Aims and Objectives ........................................................................................................ 2
   1.4 Thesis outline ................................................................................................................. 4

2. Background & Literature Review ....................................................................................... 7
   2.1 FrameNet Overview ........................................................................................................ 7
      2.1.1 Example of Frame ...................................................................................................... 9
   2.2 Requirements Elicitation Problems ............................................................................... 13
   2.3 Quality Factors of Requirements ................................................................................ 14
      2.3.1 Clarity .......................................................................................................................... 14
      2.3.2 Consistency ................................................................................................................ 15
      2.3.3 Completeness ............................................................................................................. 15
   2.4 Linguistics approaches used for requirements elicitation ............................................ 16
      2.4.1 Computational Linguistics for Requirements Elicitation .................................... 16
      2.4.2 Conceptual Modeling Through Linguistic Analysis Using LIDA ...................... 19
      2.4.3 WordNet in Requirement Engineering ................................................................. 20
   2.5 Ambiguity in Requirement Engineering ........................................................................ 22
      2.5.1 Checklist Based Inspection Approach ..................................................................... 23
      2.5.2 Controlled Natural Language .................................................................................. 25
      2.5.3 Knowledge-Based Approach .................................................................................... 27
      2.5.4 Heuristics Based Approach ....................................................................................... 29
2.6 Related Work to Use Case and Use Case Quality

2.6.1 Guiding Use Case Authoring: Results of an empirical study

2.6.2 Replicating the CREWS’ Use Case Authoring Guidelines Experiment

2.6.3 Comparing Use Case Writing Guidelines

2.6.4 Improving the Quality of Use Case Descriptions: Empirical Assessment of Writing Guidelines

2.6.5 Assessing the Quality of Use Case Descriptions

2.6.6 An Empirical Quality Assessment of Automotive Use Cases

2.6.7 Quality and Understandability of Use Case Models

2.6.8 Application of Linguistic Techniques for Use Case Analysis

2.6.9 Empirical Evaluation and Review of a Metrics–Based Approach for Use Case Verification*

2.6.10 A Pattern Language for Use Cases Specification

2.6.11 Summary

3 Approaching the Problem of Bridging Conceptual Gap

3.1 Aims

3.2 Methodology

3.3 Feasibility Study

3.3.1 Detailed Illustrative Example of Method

3.3.2 Clarity, Completeness, and Consistency of FrameNet enhanced use case

3.4 Summary

4 Towards Completeness, Clarity and Consistency

4.1 Research Question

4.2 Methodology

4.2.1 Sampling

4.2.2 Training and Material
Survey ................................................................................................................................. 116

6.1 Aims and objectives ...................................................................................................... 116

6.2 Research Questions ....................................................................................................... 117

6.3 Methodology .................................................................................................................. 118

6.3.1 Sampling .................................................................................................................... 118

6.3.2 Questionnaire & Protocol ......................................................................................... 119

6.3.3 Data collection ........................................................................................................... 120

6.3.4 Data analysis ............................................................................................................ 121

6.4 Results and Discussion ................................................................................................. 121

6.4.1 Number of Respondents: ......................................................................................... 121

6.4.2 Results of RQ1 ......................................................................................................... 122

6.4.3 Results of RQ2 ......................................................................................................... 124

6.5 Threats to validity ......................................................................................................... 130

6.5.1 Internal Validity ........................................................................................................ 130

6.5.2 External Validity ....................................................................................................... 130

6.6 Summary ....................................................................................................................... 131

7 Conclusions and Future work ......................................................................................... 132

7.1 Thesis Summary .......................................................................................................... 132

7.2 Conclusion ..................................................................................................................... 134

7.3 Future work: ................................................................................................................. 134

Appendix .............................................................................................................................. 136

References .......................................................................................................................... 163
Abbreviations

**3CS:** Completeness, Clarity and Consistency

**CBIA:** Check Based Inspection Approach

**CK:** Context Knowledge

**CNL:** Control Natural Language

**CL:** Computational Linguistics

**CO:** Concept Ontology

**CREW:** Co-operative Requirements Engineering With Scenarios

**FE** Frame Elements

**GMS** Garden Management System

**HBA** Heuristics Based approach

**HWS:** Health watcher System

**NL** Natural Language

**NLP** Natural Language Processing

**KB** Knowledge Base

**PCS:** Pre-conceptual schema
**RE**  Requirement Engineering

**UC**  Use Case

**LIDA**: Linguistic Assistant for Domain Analysis

**RQ**: Research Questions

**RSD**: Requirements Specifications Document
Chapter 1

1. Introduction

This chapter introduces the research topic of the thesis. Section 1.1 gives an overview of the research context. Section 1.2 describes the problem domain and motivation. Section 1.3 presents the overall aims and objectives of this thesis. Finally, Section 1.4 represents the thesis outline.

1.1 Overview

Requirements Elicitation is a process that encompasses activities involved in discovering and elaborating the requirements of a software system. It is considered as the most vital activity of Requirement Engineering (RE) process where system developers and engineers work in close association with customers and end-users to determine the requirements of a system [1].

Software requirements embody the goals of a software system; what functionalities should comprise, along with any ancillary non-functional properties related to performance and usability. It is intuitively important to ensure that requirements are accurate. Not only do they form the basis for software cost estimations where missed requirements can result in cost overruns, but mistakes in requirements (or misinterpretations by developers) can also lead to fault in the software itself [2].

Requirements are conventionally developed by a process such as by collating a set of informal, relatively verbose Natural Language (NL) documents and subsequently refining these into more concise, structured components such as use-cases [3]. This is an imperfect process, often resting on the intuitions of the requirements engineers, their ability to communicate with appropriate stakeholders and to home-in on the key concepts and concerns. Invariably, important considerations can easily be omitted, and the uses of NL whether structured or not can easily lend itself to misinterpretation [4, 5].
1.2 Problems and Motivation

Requirements elicitation is largely influenced by the communication skills of requirements engineers [6]. Requirements engineers work closely with customers and end-users to discover and elaborate requirements for system-to-be. Since the stakeholders involved in this activity belong to a different background, therefore the communication gap between analysts and stakeholders is well acknowledged as one of the key problem areas in requirements elicitation [7-10]. There are two crucial reasons of communication gap between analysts and stakeholders [6]. Firstly, NL specifications are prone to incompleteness, inconsistency and unclearness; and the customers often insist on using NL in requirements specification [11] as they can understand these easily. Secondly, the poor domain knowledge and divergent perceptions may lead to wrong specifications, as different groups involved in requirements elicitation may have different interpretations for the same requirements description [7]. Therefore, to elicit clear and consistent requirements, it is essential to build a common understanding of text-semantics between both, analysts and customers.

Related work on this topic has taken various directions: from formalizing specification to restricting and controlling NL; and developing domain ontologies to bridge the conceptual gap between stakeholders and developers. Though formalized languages reduce ambiguity and improve clarity but remain difficult and inconvenient for most stakeholders to understand.

The constraint languages (such as Gerkin) [12] have found a wide use within agile RE, and help to reduce ambiguity to a great extent. Yet, these do not address the problem of missing information and sometimes inconsistent domain interpretation.

1.3 Aims and Objectives

The aim of this thesis is to tackle the problem of conceptual mismatch and ambiguity in NL requirements by improving understandability of the semantics of text-based requirements in light of both linguistics and domain knowledge. The conceptual mismatch and lack of semantic agreement among stakeholders [7] can lead to poor elicitation of requirements.

As RE still lacks detailed ontology in terms of commonly agreed words, definitions and meanings [13], it is required to have a common model for concerns which is equally
understandable by all stakeholders. The proposed research builds on the work of using ontologies for establishing common understanding [14-17]. Since human cognition is a reflection of human language [18], therefore the Fillmore’s project FrameNet could serve as a stepping stone towards common model for concerns or understandings. FrameNet has previously been used as a knowledge base for NL [19, 20] and to provide a conceptual structure [21-23] for language. However, the potential of FrameNet has not yet been used in requirements elicitation activities. Since the use cases are the main medium of specifying user requirements [24, 25], our main focus is on quality enhancement of use cases.

The work proposed in this thesis enables the requirement analysts to utilise FrameNet as a knowledge base to obtain good quality use cases.

Following are the main objectives of this research:

i. To highlight the linguistics problems (shortcomings) encountered during requirement elicitation and use case development.

ii. Presenting new method for use cases elicitation using FrameNet frames.

iii. Using FrameNet frames as knowledge base for requirements elicitation to obtain relevant and clear information in different scenarios.

iv. To evaluate the completeness, consistency and clarity of use cases by utilizing different components of frames.

v. Enable requirements analyst to acquire a complete set of valid requirements by using frames.

vi. To determine the patterns followed by different users, using frames for use case development.

vii. To check that either the additional information included in use cases are considered equally useful by requirements analyst and domain expert.

To fulfill these objectives the mixed methodology approach is used, where explorative, experimental, quantitative, and qualitative study took place. The detail of how these objectives are achieved is as follows:

i. A brief literature of textual description problems in requirements elicitation and use case development due to NL is presented.
ii. A novel approach is represented for tackling the textual description problems by using FrameNet frames, which are linguistics patterns for concept description.

iii. An explorative study took place to explore that either the FrameNet plays a role of knowledge base to complement the knowledge and perception of the requirements analyst.

iv. A set of predefined metrics are used to assess the completeness, consistency and clarity of use cases developed with frame and without frame.

v. A significance difference of valid and invalid requirements is calculated between participants, elicited requirements with and without frame to test FrameNet efficiency.

vi. A think aloud approach is used for interviews to explore how different participants used frames for use case elicitation.

vii. A survey took place among software developers and domain experts to get their feedback about frames.

1.4 Thesis outline

Chapter 2

This chapter provides background information of requirements elicitation and the problems faced by analyst using NL for requirements elicitation. This chapter gives brief detail of techniques and approaches used for solving issues of NL. This chapter also gives detail of use case, use case quality and the metrics used to measure the use case quality. The detail overview of FramNet is also explained in this chapter. Finally, the guidelines and approaches, previously adopted to obtain good quality use cases are discussed.

Chapter 3

This chapter presents a novel approach for tackling the textual description problems in use cases by using FrameNet frames, which are linguistics patterns for concept description. The proposed approach is demonstrated via an explorative example of two subject system.

Chapter 4
This chapter presents a preliminary investigation on stated hypothesis holds true. To this end we conducted an experimental study, involving students working on use case elicitation. A set of metrics proposed by Torner et al. [26] is used to evaluate the consistency, completeness, and clarity of use cases.

**Chapter 5**

A qualitative study is performed to compare the difference between requirements elicited with and without frames, where interviews with 10 requirements analysts have been carried out. This study investigated the role of frame in obtaining valid and invalid number information.

Furthermore, the pattern adopted by participants using frame for use case development is also investigated to check either there is any common pattern followed for requirements elicitation using frame.

**Chapter 6**

This chapter introduces quantitative study conducted among software developer and domain expert to investigate the usefulness of frame on a large scale of experienced users. This study aims to provide an evidence, that the frame can considered as supportive tool between technical and nontechnical stakeholder to reduce communication gap.

**Chapter 7**

This chapter presents conclusions and further work.
CHAPTER 1: INTRODUCTION

CHAPTER 2: BACKGROUND & LITERATURE

CHAPTER 3: METHODOLOGY

CHAPTER 4: EXPERIMENTS

CHAPTER 5: THINK ALOUD INTERVIEW

CHAPTER 6: SURVEY

CHAPTER 7: CONCLUSION & FUTURE WORK

Figure 1.4-1 Thesis Structure
Chapter 2

2 Background & Literature Review

This chapter provides background information of Requirements Engineering (RE) and is mainly split into four sections. The first section demonstrates an overview to FrameNet frames [27]. The second section provides an overview of the requirements elicitation and highlights the effects of communication barriers among stakeholder on requirements elicitation. The third section comprises of the approaches used to deal with the requirements elicitation problems. The fourth section discusses the use case overview, quality of use case, guideline used to improve the quality of use case and the measurement of use cases.

All the techniques that we survey are discussed in terms of three core attributes: Clarity, Completeness and Consistency (3Cs). The term Clarity refers that how easily the requirements are understandable by stakeholders. The requirements are Complete if they are enough to establish the goal set by the customer. Consistency is related to the absence of conflict among requirements. The detailed discussion of 3Cs is available section 2.3.

These are essential attributes for any RE technique [28], and will form the basis upon which we develop the FrameNet-based approach covered in this thesis.

2.1 FrameNet Overview

In NL (such as unrestricted English) linguistics, a single word can be ambiguous because of homonyms, synonyms and polysemy, which can make hard to fully understand the semantics of a piece of text [27]. In order to overcome these problems, the Berkeley FrameNet project was developed by Fillmore [29-34]. The real aim behind development of FrameNet is to understand the syntactic and semantic values of a word, as one cannot fully understand the meaning of a single word without having access to other essential contextual knowledge [35]. Frames are the “unified schematization of experience” where the meaning of a word is the outcome of relationship between the word and frame semantics. Word meanings become less ambiguous by considering all relevant concepts with in a frame.
The FrameNet project can be considered as an online dictionary that gives not only the meaning of a word but reveals the related contextual information to word. FrameNet has the capacity to link certain words to their meanings and also provides these definitions in the context of “frames – relationships” to other terms that serve to show how the words are used, and what other terms/concepts are required to derive their ultimate meaning.

Work on frame semantics has been applied to an English language corpus by the FrameNet project. The current database comprises approximately 13,640 words and definitions (known as Lexical Units (LUs)) and 1,224 frames [36]. These words are supported by corpus evidence and documented with examples. The actual process of LU identification and frame construction is beyond the scope of this thesis and is detailed in [37].

Frame formation starts with linguists reviewing use of a given word (e.g., 'book') to identify the patterns of its use, i.e., what other words frequently used with it (e.g., booker, organization, schedule-time, client, etc.), what semantic it provides (e.g., booking often implies reserving of an activity by some party to someone) and what syntactic it implies (e.g., book is normally the verb with the booker acting as the subject and an activity booked from the other party as an object) and structures does that word and other words around it bring up [27, 31]. When a set of words (e.g., booker, services, book, schedule_time, organisation etc.) coalesce together in expressing the meaning of a particular concern, they are considered core Frame Elements (FEs) (or lexical units) of the pattern. The initially identified patterns are then combined into frames. Within a frame, each syntactic structure is assigned a clearly defined semantic role (e.g., the FE Booker describes the role of a person or organization that books an activity). Each frame is validated against a set of rules that defines its completeness against reviewed examples and relations to the previously defined frames. A sample rule, for instance, requires that all examples included in a single frame cover all its core lexical unit [31].

Frames are also processed through reframing process. The frequent work and usage of FrameNet gave rise to the need to refine FrameNet frames. This process is known as “reframing the frame” [38, 39]. The reframing process is explained with an example Make_noise. At beginning there was only one frame for word noise i.e Make_noise. By reanalysis further frames are discovered i.e. Cause_to_make_noise, Sound_movement,
and Sounds. Two more types of frame to frame relation have also been introduced via reframing i.e. Causative_of and Inchoative_of [38].

Over the past couple of decades, the FrameNet project [30, 40-42] has been developed within the NLP community to support a raft of activities that involve attributing ‘meaning’ to NL text. FrameNet project also plays a role of knowledge base for NL processing [32, 43, 44].

Barzdins Guntis [20] introduced an approach named CNL (control natural language)-FrameNet where FrameNet is used to extract information from NL text. The framework is used on NL documents and can be used to obtain a tailor-made frame-ontology knowledge that helps in producing a simple FrameNet-CNL para-phrase text in multiple languages. FrameNet alone does not define Abstract Knowledge Representation (AKR) but is used with entity identification procedures to generate a usable AKR framework. The proposed framework is used by national news agency in Latvia to extract the biographical data profiles of publicly visible persons and organizations mentioned in the newswire articles. A Frame-semantic parser was later utilized as a national parser by the news agency for quality evaluation [20].

Berzins, V et al. [45] suggested that the use of Corpora (FrameNet) with NLP can enhance semantics understanding and can solve the problems of NLP such as communication gap among stakeholder, ambiguous nature of NL, tacit knowledge. But they have not provided any methodology that how FrameNet can be utilized to elicit requirements.

To understand the FrameNet, it is necessary to understand the basic components of it. The basic FrameNet components are Frames, FEs, LUs, Frame-to-Frame relation, Valances and Annotation. The components of frames are illustrated (see figure2.1-1) below with an example of “Reserving” frame.

### 2.1.1 Example of Frame

The word “book” is used here to explain the different components of FrameNet frames. The word “book” has two different meanings: book as a noun is a collection of texts, while book as a verb stands for making a reservation. In FrameNet, these two concepts are described by two different frames “Text” and “Reserving” respectively. The words that evoke a given frame are the frame’s LUs.
To understand the concept of booking or reservation, one has to realize/know that there is a facility that is booked or to be booked. The detail of “Reserving” frame can be checked on FrameNet frames website [46].

2.1.1.1 Reserving Frame Definition

In this frame, “…a Booker brings about a situation where an Organization plans to provide Services to a Client at a particular Scheduled_time. The Organization has authority over when Services are provided to whom. The Booker usually has to communicate and negotiate with the Organization to plan the Services for a mutually agreeable Scheduled_time. Usually, there is a Payment as well” [46].

Core FEs

Booker: The entity that contacts the Organization directly. In many if not most cases, a Booker is also a Client but the roles may also be different, e.g. a travel agent.

Client: The entity that is intended to receive the Services of the Organization.

Organization: The entity that enters the request of the Client into its records to retain Services in the future.

Scheduled_Time: The future point in time when the Client arranges for the Organization to provide Services.

Services: The goods or labor that the Client requests the Organization preserve for future consumption or rendering.

None Core FEs

Manner: Any description of the event as a whole ("the same way") or a characteristic of the Booker during the action that holistically affects the event ("sloppily").

Means: An action or methodology used by the Booker to bring the reservation about.

Payment: The amount paid to the Organization in exchange for the Services.

Place: The location the Booker performs the action.

Purpose: The situation that the Booker intends to bring about by reserving.

Time: The time at which the reserving event takes place.

I booked the room for Monday only on Tuesday.
Figure 2.1.1-1 illustrates the schematic representation of Reservation frame where blue circles represent core FEs and yellow circles represent none core FEs.

![Figure 2.1-1 Schematic representation of “Reserving” Frame](image)

**Lexical Units (LUs)**

A **LU** is the combination of a lemma (i.e., base form) of the given word with its meaning. Every new meaning of a given word is presented by a new LU and where each LU evokes a new frame (e.g., book a ticket vs. read a book). On the other hand, words with same meaning can evoke same frame (e.g., reserve a ticket and book a ticket). Thus, the problem of lexical ambiguity in terms of synonyms and polysemy can be resolved through LU. The synonyms and polysemy for the word “book” are described in figure 2.1-2

![Figure 2.1-2 Lexical Unit: polysemy and homonyms of reserving](image)
**Frame Elements (FEs)**

A frame comprises a set of slots of semantic roles which are known as *frame elements*. The FEs are used to label the different parts of a frame. The conceptual integrity of the frame is defined by the interrelation between the FEs which is best illustrated with the following example:

The “Reservation” frame must have a *Booker* (i.e., an entity that carries out reservation), *Organization* (the organization where the reservation is made), *Scheduled_time* (time of the reservation), *Services* (the reserved services), and *Client* (the entity for which the reservation is made): E.g., Aaron (Booker) booked tickets for Ann and Steve (Client) to see the Romeo and Juliette (Service) at Grand Theatre (Organization) at 19.00 tomorrow (Scheduled_time).

The set of FEs which must be filled in order to provide full information expected from a given frame are called the *core elements* of the frame [20]. In the above example all FEs are core to the “Reservation” frame, if any of core FEs is omitted, the reader/listener will be not fully informed about the given reservation.

Additional information for the frame can be provided through its non-core elements which, for the “Reservation” frame are *manners* (e.g., Arron hastily booked), *means* (booked via computer….), *place, payment, purpose, and time*.

**Frame-to-Frame relationships**

Another important feature of FrameNet project is that frames are arranged into an ontology using *Frame-to-Frame relationships* [23]. The frequently utilized relations are as follows:

‘*Inheritance*’ It is like a parent/child hereditary relationship where the child frame is a subtype of parent frame and each FE in parent results into a similar FE in the child. The FrameNet project speaks of frame inheritance relation in the form of inheritance by and *inheritance from*. The *Reserving* frame is inherited from *Schedule* frame.

‘*Using*’ is a relationship between parent and child where child uses parent frame as a background though it is not necessary to have each and every FE of the parent to be bound to a similar FE in the child. [47]. The *Reserving* frame is not ‘using’ any frame.

‘*Sub-frame*’ relation where several smaller sub-frames comprise parts of a larger complex frame [22]. The *Reserving* frame has no sub-frame.
Valences

These are words which need other words in order to produce a semantical meaning. For instance, the independent use of a verb in the absence of subject and object makes no sense. Verbs always need some arguments in the form of nouns and pronouns. A particular word/phrase combination needed in a sentence is termed as the valence of the word. In FrameNet frames a single word (FE) is grouped with other words (FEs) to produce semantics meanings.

Annotations

In FrameNet frames, the parts of sentence are annotated to represent FEs. These annotations were used to identify the specific FEs in each sentence. For example: The “Reserving” frame FEs are annotated with different colors.

“a Booker brings about a situation where an Organization plans to provide Services to a Client at a particular Scheduled_time. The Organization has authority over when Services are provided to whom.” [46].

2.2 Requirements Elicitation Problems

NL texts can pose a range of challenges when used for requirements elicitation [48]. Some of the core problems can be summarized as follows:

- The ambiguous nature of NL leads to conceptual mismatches amongst stakeholders because the ambiguous sentences can be easily misinterpreted and overlooked by humans.
- The understanding of NL is highly dependent on the specification of language i-e the readers and writers must use same words for the same concept.
- Requirements specified in NL are overly flexible as similar requirements can be expressed in completely different ways due to ambiguous nature of NL.
- Modularization of NL requirements is another big issue as there is no easy mode to identify all interrelated requirements. To discover the consequence of a change, one has to look at every requirement rather than just a group of related requirements.
Relying only on NL to elicit requirements without using any other additional tools may result in incomplete, inconsistent, ambiguous, unclear and unstable user requirements. Therefore, it is necessary to establish a common understanding of text-semantics by both analysts and customers to obtain good quality requirements.

2.3 Quality Factors of Requirements

There are many attributes which can affect the quality of requirements such as Cohesiveness, Completeness, Consistency, Correctness, Clarity, Lack of Ambiguity, Usability, Validatability, Verifiability and Feasibility [49]. In current research Clarity, Completeness and Consistency (3Cs) [28] are selected to assess the quality of use cases. Since in current research the focus is to enhance the quality of use cases (using lexical) database therefore the basic quality attributes of use case are selected. The 3Cs are recognized as key quality properties of use cases and they have been commonly adopted by the related research [50] [51] [26] [52] [53]. These properties have causal relationships between each other. That is, improving one property should not reduce another property. All these properties should be adequately maintained in order to have good quality of use cases.

2.3.1 Clarity

Clarity is considered as one of the most fundamental condition for good quality requirements. The term ‘requirement clarity’ refers to that how easily the requirements are understandable by stakeholders. Unambiguity and semantics understanding of requirements is key step to obtain clearer requirements [54]. The clarity of use case is measured in terms of readability, clear structure of main/alliterative steps and semantic conflicts within the use cases [26] [52].

2.3.1.1 Semantics understanding and control to ambiguity

The clarity of requirements is largely dependent on semantics understanding and unambiguity of language used for requirements elicitation. In RE, it is important to have clear semantics understanding of situation before starting the process of requirements elicitation [6], as people involved in RE belong from different backgrounds and can interpret the requirements in more than meaning. The semantic
understanding of requirements is also essential to determine the conflict among requirements as requirements interact via their semantics.

2.3.2 Consistency

The term ‘consistency’ refers to situations where a specification contains no internal contradictions among requirements [28]. In RE, the requirements can be interrelated and interdependent to achieve the goal [25].

2.3.2.1 Interrelated requirements

Interdependencies among requirements can easily lead to inconsistency, as the satisfaction of one requirement depends on the satisfaction of another [55]. Inconsistency in requirements also occurs due to requirements clashes or bad dependencies, where completion of one requirement disturb completion of another requirement. To overcome the problem of inconsistency, the reader must understand the semantic relationships between requirements [56].

2.3.2.2 Modularity of concerns

A concern is a property of interest to stakeholders; as such the requirements of stakeholder are concerns [57]. A concern can be defined as a unit encapsulating one or more requirements related to a certain matter of interest such as a use case. A concern can be simple (containing only requirements) or composite (containing other concerns). Composition and decomposition of concerns are known as modularity of concerns. The principle modularity of concern divides larger problems into small components and deals with each component separately. There are a number of advantages of using this principle, such as reducing software development complexity by dealing with each issue separately, division of efforts and improvement of systems’ modularity [58]. In RE the term modularity of concerns refers to compose requirements in a meaningful way [59].

2.3.3 Completeness

Completeness is an essential quality attribute of RE ensuring that all relevant requirements are captured in a specification. According to [60], requirements are complete if they are sufficient to establish the goal they are defining.
Completeness is mainly dependent on relevant information of a particular concept. The cause of incomplete requirements is mostly the missing information and a lack of common understanding amongst the stakeholders [55, 56, 61].

2.4 Linguistics approaches used for requirements elicitation

Computational linguistics is a set of tools and methods to support linguistic investigation across all branches. It is an interdisciplinary field dealing with the statistical and/or rule-based modelling of NL from a computational perspective.

Linguistic-based methods and tools can be considered as appropriate tools for RE as they are designed to capture NL descriptions of particular domains [62, 63]. Bures highlighted the importance of linguistic tools and suggested that the use of linguistic techniques may perform a crucial role in providing support for requirements analysis.

Linguistic-based ontologies can be glossaries, dictionaries, controlled vocabularies, taxonomies, folksonomies, thesauri, or lexical databases [14]. These introduce a common understanding to reduce misinformation among stakeholders in RE. There follows a review of key linguistic tools that are used to elicit requirements.

2.4.1 Computational Linguistics for Requirements Elicitation

Computational Linguistics (CL) is a set of techniques to process words and texts in NL. Computational understanding of a language provides insight into thinking and intelligence as language is a mirror of mind [64].

Zapata [65] conducted a survey on related work to CL techniques used for software development automation. The author summarized some of the efforts invested in supporting the requirements elicitation process using CL techniques. The author also highlighted that the shortfalls of requirements elicitation in terms of NL can be solved by using CL and Natural Language Processing (NLP) techniques. The CL tools1 related to current research are discussed as follows:

1 Tools related to current study are selected, those which are used for Spanish language are ignored
2.4.1.1 Pre-conceptual schema (PCS)

The PCS is a graphically representation of previous knowledge about a concept. The PCS are used to generate conceptual diagrams automatically for specific domain. In PCS, nouns from stakeholder’s discourse are treated as a concept. Discourse is the stakeholder’s requirements written in control language [66].

The PCS is mainly developed by an analyst during requirements elicitation, to provide visual support to stakeholders such as UML (unified markup language) diagram. Since the UML are understandable by technical people only, PCS are introduced to get the graphical language closer to the stakeholder discourse. Although the PCS provides a new approach to clarify a customer’s views about a domain, the usage of PCS is mostly limited to analyst and least supportive for the customer.

2.4.1.2 UN-Lencep: specifying pre-conceptual schemas

UN-Lencep language (acronym of a Spanish phrase for National University of Colombia Controlled Language for Pre-conceptual Schema Specification) is a subset of NL, known as Control Natural Language (CNL). UN-Lencep simplifies automatic obtaining of PCS from stakeholder’s discourse [66]. UN-Lencep is introduced to overcome the problem of a pre-conceptual schema by involving both the customer and developer. The UN-Lencep discourse is manually created by the analyst in collaboration with the customer [65].

2.4.1.3 Dialog Model (DM)

The UN-Lencep language overcomes the drawback of PCS, but Zapata [65] still raised the need of developing, a more customer friendly UN-Lencep language. As the requirements elicitation started with set of interviews between stakeholder and analyst, thus they emphasised to obtain the UN-Lencep discourse from something like an interview [65]. The interviews are dialogues between analysts and stakeholders, the author proposed to use DM for developing UN-Lencep language from interviews. He proposed a structure (figure 2.4.1-1) for requirements elicitation dialog.
2.4.1.4 UNC Corpus

Modelling is one of the important procedure trends in requirements elicitation. Modelling is helpful to enhance the understanding of a domain as models illustrate relationships among different concepts of a particular domain. Usually, models are developed by requirements analysts (developer) using their experience and domain knowledge. “A corpus is a collection of texts used for linguistic analyses usually stored in an electronic database so that the data can be accessed easily by means of a computer Trend”.

2.4.1.5 Analysis of CL with the 3Cs

- Clarity

  Control of ambiguity: CL tools provide support to control ambiguity, particularly the homonymous lexemes normally produce distinct entries in dictionaries, where linguistic analysers automatically resolve the homonymy by an appropriate selection of the available options in the dictionary. Although CL tools can be helpful to control ambiguity but none of them provides an approach like FrameNet where all the relevant concepts (not only meanings of a word) are explained by individual frames.

  Semantics understanding: CL tools do support the task of understand the meaning of a word and provide schemes for a single word, but they fail to provide interrelated concerns of a word which can be helpful in developing any concept.

- Completeness
CL tools do not give explicit detailed information about specific concepts, which can be helpful to highlight missing information. Thus, CL tools can clarify the customer textual requirements but do not provide any support to determine missing requirements as the FEs do. In FrameNet frames, an absolute idea about a single word is expressed via the FEs and frame to frame relation.

- **Consistency**

  **Interrelated requirements**: CL tools do not provide any support to identify interrelated requirements because they do not offer any detailed information of particular concept.

  **Modularity of concern**: CL tools provide support in modularity of concern especially in terms of pre-conceptual schemas. In pre-conceptual schemas the visual support is provided to stakeholder by presenting a sequence of text in graphical format.

### 2.4.2 Conceptual Modeling Through Linguistic Analysis Using LIDA

The authors in [67] pointed out that although Object Oriented (OO) modelling techniques offer a progressive change in software development, but that the problem of identifying the model element types such as classes, attributes and methods, still remains. For this the author proposed a methodology and a prototype tool, Linguistic Assistant for Domain Analysis (LIDA) to generate conceptual modelling or OO models through linguistic analysis. In LIDA, the textual description of requirements is transformed into OO concepts.

LIDA carries out this analysis by compiling a word list and multi-word terms contained in a document. It also presents an interface for users to identify them against the corresponding elements of a Model. LIDA has two main sections; the LIDA Text Analyzing Environment and the LIDA Model Editing Environment.

In the Text Analysing Environment section, the text is first read and then Part Of Speech (POS) [68] [31] segments are assigned to words. A word count is performed on base form of words and finally a user assigns the model element type (such as class or attributes) to words or phrases.

In the LIDA Model Editing Environment, the candidate’s model elements marked in LIDA’s Text Analysing Environment are converted into model form (i.e. UML Class model). The textual description of the model is also generated to document and validated with domain experts.
2.4.2.1 Analysis of LIDA with 3Cs

- Clarity

**Control of ambiguity:** LIDA contains the morphological knowledge of English which structures the word but does not give the details of synonyms and homonyms. Thus, we found that LIDA is not particularly supportive for controlling ambiguity.

**Semantics understanding:** LIDA extends a good integration between the text analysis process and the conceptual modeling process and comprises of morphological awareness of English but it does not give semantic knowledge, therefore, we can say that LIDA is not supportive of semantics understanding.

- Completeness

LIDA utilizes the existing text description of the problem domain and the initial class diagram with attribute, methods and roles is produced from them but it does not give any support in identification of missing information which helps in obtaining complete requirements.

- Consistency

**Dependent requirements:** LIDA provides no support to identify interrelated information.

**Modularity of concern:** LIDA is helpful in obtaining UML models from textual descriptions but it does not provide support to modularity of concern by using LIDA.

2.4.3 WordNet in Requirement Engineering

**WordNet** is a large lexical database of English developed at the Cognitive Science Laboratory where nouns, verbs, adjectives and adverbs are grouped into sets of synonyms called synsets. Each synset represents a distinct concept, general definitions, and also records the various semantic relations between these synonym sets.

A prominent difference between the WordNet and a standard dictionary is that WordNet distributes the lexicon into five divisions, namely nouns, verbs, adjectives, adverbs, and function words. WordNet is also used in RE by many researchers to obtain unambiguous requirements as discussed below.
Wolter, K et al [69] introduced a Requirements Specification Language (RSL) to obtain unambiguous requirements by utilizing WordNet [70] glossier. RSL provides a relationship between requirements specifications, domain specifications and the global terminology. The global terminology has a connection with WordNet and it consists of different words with same meaning from different domains and then these words are linked with WordNet. Linking words as WordNet is a semi-automatic approach performed by RSL editor tool.

Matsuoka, J. and Y. Lepage [16] also used WordNet to check ambiguities in Software Requirements Specification texts and to identify ambiguous words in sentences. This helps a human reader to make a decision about the correct use of the words or terms according to a particular situation. The author introduced a resource and three techniques for detecting ambiguous terms. Two techniques were knowledge-based methods while the one (C-value) was a linguistic and mathematical method used to extract technical words.

Li, R., K. et al. [71] pointed that to obtain clear semantics understanding of requirements, it is necessary to deal with word semantics and sentence semantics separately. In this regard, a requirement ontology is presented which uses two techniques: Generalized Upper Model (GUM) and WordNet to deal with both sentence and word semantics respectively. The first technique (GUM) is a general task and domain independent linguistically motivated ontology' that provides semantics for natural language expressions. The GUM has two hierarchies. First one contains all the concepts and the second level contains all the roles. In this paper, the first hierarchy (i.e. conceptual part) of GMU is used to understand the semantics of a sentence. The second technique (WordNet) is used to control lexical ambiguity by obtaining different meanings of a word.

2.4.3.1 Analysis of WordNet with 3Cs

- Clarity

Control of ambiguity: WordNet can serve to control ambiguity as it provides synonyms and polysemy of a single word along with definitions. The lexical ambiguity in terms of synonyms and polysemy is tackled with Lexical Units of FrameNet frames.
Semantics understanding: WordNet supports the understanding of the semantics of a single word only, as WordNet grouped the English words into a set of synonyms called synsets.

- Completeness

WordNet provides support to semantic understanding of a single word but it does not provide relative information about single word. Semantic understanding can be helpful to determine relevant information but still it cannot be much helpful to develop conceptual modelling. Thus, we can say that WordNet does not provide any support to obtain missing information. The FrameNet, on the other hand, describes the Frames to Frame relation and Frame Elements illustrate all relevant ideas about single word.

- Consistency

Interrelated requirements: It just provides the different meanings of a single word. Since it lacks relevant information about the word, it seems that WordNet provides no support to determine interrelated requirements.

Modularity of concerns: It does not provide any support for modularity of concern as WordNet just groups words into synonyms. WordNet does not define all the role of a concept as done in FrameNet frame via frame elements.

2.5 Ambiguity in Requirement Engineering

The importance of NL in RE cannot be ignored [47, 72, 73], as the customers do not sign the project if the requirements are written in any language other than NL [11]. Sommerville [74] identified ambiguity as one of the main limitations of NL in the behavioral specification of a system. He further elaborated that ambiguity in terms of synonyms and homonyms leads to implicate and explicit assumption of stakeholders.

According to Anderman and Rogers [75], ambiguity arises from linguistics and conceptual differences amongst communicators, where a human often overlooks the ambiguity and misunderstands the real meaning of ambiguous sentences.
The requirements are considered ambiguous if they are interpreted in more than one meaning and lead to more than one direction [76].

Shah and Jinwalain [77] carried out a literature survey identifying lack of information and communication error as two major causes of ambiguity. They differentiate two broad categories of ambiguities in terms of linguistic ambiguities and software engineering ambiguities.

Linguistic ambiguities can be raised in different forms such as lexical ambiguity, semantic ambiguity, Syntactic ambiguity, Referential ambiguity and Pragmatic ambiguity. A detailed systemic review of the literature has been done by [78] to address the challenges of ambiguity in RE. The result of her study described that the main emphasis of most of the studies is on detection of ambiguity rather than reducing ambiguity, where only 4-5% of work has performed to reduce ambiguity. The previous work has largely focused on syntactic ambiguity detection rather than lexical and semantic ambiguity.

The author further described the tools used by various researchers for addressing ambiguity where most of the tools are common, as used in [77]. The main approaches used so far to detect ambiguity are Check List Based Inspection, Control Language, Style Guides, Knowledge-based and Heuristics-Based approaches [77, 78].

Ferrari et al. elaborated in [79] that ambiguity is the major barrier among communicators while transferring knowledge. In the view of the fact that ambiguous requirements are often seen as a cause for project failure, relying only on NL to elicit requirements without using any other tools may leads to several problems like incompleteness, inconsistency, and unclear and unstable user requirements.

There follow some of the approaches, tools and techniques which have been used in previous research to deal with ambiguity in RE. Some of the approaches which deal with ambiguity have already been discussed previously with respect to CL.

### 2.5.1 Checklist Based Inspection Approach

The Checklist based inspection approach (CBIA) is one of the most common and popular approach used in dealing with lexical ambiguity. The CBIA was previously used by [80-82] to identify ambiguities in RE. The most successful approach of CBIA is to hand over the requirements to different stakeholders and each stakeholder is asked
to interpret the requirements. The interpretations of different stakeholders are compared and if the requirements differ, the requirements are considered ambiguous.

Kamsties, E.et al. [81] categorized the RE ambiguity separately than the linguistics ambiguity. *RE-specific ambiguities* occur with respect to RE context, which includes the application domain and the system domain.

CBIA is feasible for a small set of requirements but less efficient to a larger scale of requirements. The checklists-based reading offers support to identify ambiguity but sometimes it fails to deal with complicated types of ambiguities. Therefore, scenario-based approach is used in combination with CBIA to control ambiguity.

In scenario-based reading, an inspector is provided with an operational scenario where initially the product abstraction of the product (i.e. the requirements document in our case) is created by the inspector. Then the inspector is asked to answer the questions according to the analysis of the abstraction which emphasizes on the role that the inspector assumes.

The checklist and scenario-based approach depends on the particular RE context in which the RE process takes place.

2.5.1.1 Analysis of CBIA with 3Cs

- **Clarity**

  *Control of ambiguity*: The CBIA is specifically proposed to control ambiguity in RE. It should be noticed that CBIA does not provide control to domain specific ambiguity in general but the FrameNet frame is a generalized approach.

  *Semantics understanding*: It seems that CBIA is helpful in semantic understating as it clarifies the meaning of ambiguous information.

- **Completeness**

  The checklist approach provides support to deal with ambiguity in specific domain of RE but it does not provide any explicit support to completeness of requirements. Although the semantic understanding of the requirements provides a clear perceptive and this can be helpful to recall relevant information, but we cannot say that checklist provides any support to retrieve any relevant missing information. Thus, the checklist-based approach does not entirely support to obtain complete set of information.
• Consistency

**Dependent requirement:** Since the CBIA does not provide any support to retrieve relevant information, it therefore does not provide any support to identify interrelated information.

**Modularity of concern:** Though the CBIA provides a check to ambiguity; it does not provide any support to modularity of concern as FrameNet frame does.

### 2.5.2 Controlled Natural Language

Control Natural Language (CNL) is a restricted form of NL. The restriction on NL is imposed with respect to its lexicon and grammar in order to control ambiguity.

The grammatical restrictions minimize the complexity and ambiguity of the sentences. The lexical restrictions minimize the size of the vocabulary and the meaning of the lexical entries for a particular application domain.

Thus, the restricted form of NL can be effective to control ambiguity in RE but it reduces the expressiveness of requirements too as it only allows certain formulations, phrases, and a restricted vocabulary only.

CNL is previously used to reduce the ambiguities in RE where different researchers have discussed different specification of controlling NL as per their needs [83-85].

Reynaldo[85] used CNL to automatically generate object models (class diagram) from textual requirements. Since the NLP tools make it easy to identify classes in textual requirements but due to ambiguous text, the unnecessary classes are created too.

To reduce ambiguity in textual description of requirements, the author restricted the NL by prohibiting the use of pronoun, compound sentence and the passive form of sentences. The restricted NL potentially solved the issue of generating unnecessary classes from textual requirement, but the author pointed that still there is need to work on the detection of missing object in a sentence.

Fockel, M. and Hotmann.J in [83] proposed a model-driven RE approach to bridge the gap between requirements models and NL. In this approach, the advantages of both model-based and NL-based documentation are combined together. The CNL is also used to accompany with requirements models for an easy automatic processing of the textual requirements.
Schwitter in [86] introduced a computer-processable CNL (PENG) which is designed to write an unambiguous specification. In PENG, the sentence structures is simplified and standardized by controlling the grammar of language while unnecessary linguistic variations are removed by restricting the lexicon.

Boyd.S et al [84] argued that using CNL in RE can help to control ambiguity but it also has some side effects on the expression of the information. CNL has tendency to reduce the expressiveness of NL. The expressiveness is the measurement of different lexical and grammatical constructions allowed by CNL to use.

In this regard, the author proposed an approach to measure the expressiveness of CNL while using it in specific domain (such as they selected corpus for analysis from Australian naval defense industry). The key objective of this work is to determine whether the lexicon in CNL is sufficient to express the requirements of selected corpus Australian naval defense industry? How is sufficiency (with respect to the expressiveness of a CNL) even defined?"

The result of the work shows that each lexical entity of the CNL can be made on the overall expressiveness of the CNL. The author claimed that the proposed approach is valuable for designing new CNLs or reform existing CNLs.

2.5.2.1 Analysis of CNL with 3Cs

- Clarity

  Control of ambiguity: CNL provides support to control ambiguity by implanting restrictions on NL and allow only specific form of words and phrases to use for requirement elicitation.

  Semantics understanding: CNL is supportive in semantic understanding to some extent as CNL restricts the use of vocabulary to a domain but we cannot say that it gives a general understanding of some text as FrameNet frames do where different ambiguous words are listed in LU and each word triggered individual frame.

- Completeness

  The CNL has restricted the expressiveness of NL which can results in the overlooking of the relevant information. Thus, we can say that the CNL does not provide any extra information and support to identify missing information.

- Consistency
**Dependent requirement:** The CNL has not provided any support to retrieve relevant information, thus we can say that it does not have any support to identify interrelated/dependent requirement.

**Modularity of concern:** The CNL restricts the use of the NL as it uses some standard format of phrases, control grammar (for example using either active voice or passive voice) and restricts vocabulary but it does not give any support to modularity of concern.

### 2.5.3 Knowledge-Based Approach

The communication barrier between the analysts and the stakeholders is caused by gap in their domain knowledge [87]. Another rising cause of ambiguity in RE is the conceptual mismatch among stakeholders due to poor domain knowledge. These varying interpretations of different groups towards the requirements’ description may lead to wrong information.

Macaulay [88] identified the lack of knowledge and common understanding as one of the major causes of misinterpretation of information exchanged among stakeholders. In this context, the Knowledge Based (KB) approach is introduced to resolve ambiguities by providing extensive information about the specific domain. KB is an online library of information that stores information about domain.

Al-Harbi, et al [89] introduced an approach where concepts ontology and context knowledge (CKCO)(COCK) of a domain is used to resolve lexical semantic ambiguity in Natural Language Questioning (NLQ).

The context knowledge identifies the correct sense of words in the text using WordNet with domain labels, whereas the concept knowledge identifies the concept and then relates the concepts using ontology.

The CKCO approach is applied on university domain Question Answer (QA) system. This system can automatically answer English NLQs that are asked by new students about the university. The results of the study have showed that the proposed approach is capable of resolving ambiguous words in the NLQ which consists of multi words.

Ferrari.A et al [90] analyzed the role of ambiguity in interviews. Interviews are one of the most common approach used by stakeholders to elicit requirements. In this paper, the ambiguity of the interview is classified into four sub-phenomenon: unclarity,
multiple understanding, incorrect disambiguation and correct disambiguation. A set of 34 customer–analyst interviews is carried out to underline the different types of ambiguities and various causes of these ambiguities.

During this research, the author has explored a unique relation between ambiguity and tacit knowledge in interviews. The tacit knowledge is the knowledge that is held by the customers without sharing it with the analyst for some reasons. This has been identified as a critical cause of ambiguities.

In their interviews, no graphical representation has been provided and the analysts have recommended that any missing information about the domain could have been clearly identified with the inclusion of simple graphics. Therefore, the author raised the need of domain and context knowledge to deal with tacit knowledge in interview process.

Hadar, I. et al in their empirical study [87] examined the actual and perceived effects of prior domain knowledge on requirement elicitation through interviews. The study was carried out considering the following two research questions.

1) What are the positive and negative effects of domain knowledge on requirements elicitation via interviews, as perceived by analysts with and without domain knowledge?
2) Is there any difference in the actual interview conducted by analysts with and without domain knowledge?

This was achieved by investigating that how analysts could perceive this effect with or without having the DK. The result of the study indicated that the DK has both positive and negative impacts on requirement elicitation. The prior DK might mislead while solving new situations.

2.5.3.1 Analysis of KB with 3Cs

- Clarity

*Control of ambiguity*: The KB is supportive to control ambiguity by providing relevant information about a particular text or word. WordNet is one of the example of KB [91] which is used to deal with ambiguous words and word sense.

*Semantics understanding*: The KB is supportive in semantics understanding as it provides relevant information of a specific context but it is also pointed in [92] that
lack of common vocabulary leads the stakeholder in different direction of understanding.

- **Completeness**

The KB provides support to identify missing and relevant information in a specific context and is also supportive to obtain a complete set of requirements if it is rich in particular context. Since due to previous domain knowledge, the analyst sometimes ignores the customer’s answer as they consider that they know the answer better than customer[87], therefore KB sometime becomes a cause of incomplete requirements.

- **Consistency**

  **Dependent requirements:** The KB provide relevant information of a specific concern, therefore, we can say that KB can be supportive to determine interrelated information.

  **Modularity of concern:** The KB can be helpful to provide background or related information about a particular concern or concept, but we cannot say that it provides any direct support to the modularity of concerns.

2.5.4 **Heuristics Based Approach**

Heuristics Based approach (HBA) defines some rules to obtain immediate goal which may not be perfect but satisfactory. The HBA is also used to resolve ambiguities in requirement engineering.

Chantree, F., et al [93] introduced an approach to automatically identify coordination ambiguity. Coordination ambiguity appears when words like And/Or are used.

In this study, the actual requirements containing coordination ambiguity are obtained from corpus of requirements documents. The requirements documents are selected from different domains such as mechanical engineering, human computer interaction and telecommunications.

The study has been conducted by locating 639 sentences consisting of multiple structure coordination ambiguities with "and" or "or" in RE corpus. Regardless of considering the preference of any reading, sentences have been included in the survey, such as avoiding discrimination is essential in order to develop a realistic data for training and testing purposes.
Sentences containing multiple coordination ambiguities are divided into separate sentences that resulted in 138 sentences. Each of these 138 sentences contain a single multiple structure coordination ambiguity and each sentence could represent a unique requirement.

To determine the nature and types of ambiguity, the selected corpus was presented to 17 judges in 4 separate surveys. The judges were selected from different background like computing professionals, comprising developers, academics and research background. The 138 requirements, together with the judgments on them, form the dataset (ambiguous sentence along with human judgment).

The heuristic was then used to automatically replicate these judgments. The heuristic remove those ambiguities which can be easily interpreted by a human being and left the harmful ambiguities to be analyzed and rewritten manually.

Yang, H., et al. [94] presented a model built on machine learning architecture to automatically identify nocuous ambiguity. First, the human judgments were observed about nocuous ambiguity by surveying participants and asking them to interpret the text. The text was considered as nocuous ambiguous if interpreted in different ways by readers. The second step was heuristics that models human judgments. Heuristics consists of factors that may support a particular interpretation. Third step was building machine learning architecture. The machine learning architecture learns from a set of heuristics, each of which predicts a factor that may lead a reader to favor a particular interpretation. The heuristics are trained by using collected human judgments about ambiguity. The results of the study have showed that the proposed technique is helpful in determining nocuous ambiguity in text.

2.5.4.1 Analysis of HBA with 3Cs

- Clarity

*Control on ambiguity:* HBA provides support to control ambiguity as the major process of HBA is to determine ambiguous sentences.

*Semantics understanding:* HBA can improve semantics understating of any textual description by clearing ambiguities but it still lacks to present detailed meaning of the individual word as Frame's LU does. Thus, we can say that HBA provides moderate support to obtain clarity.
• **Completeness**

HBA provide little support to identify missing information by removing ambiguities from unclear sentences but still we cannot say that HBA provide relevant information about particular concept.

• **Consistency**

*Interrelated requirements:* HBA does not provide any support to determine interrelated information.

*Modularity of concern:* HBA does not provide any support to determine interrelated information, therefore there is no clue to obtain modularity of concern by using HBA.

### 2.6 Related Work to Use Case and Use Case Quality

The use case modeling is a most widely used approach to specify system's requirements [1]. The use cases are text documents written in plain English and considered as an integral part of requirements elicitation. Use cases are effectively used to represent systems’ functional requirements in a way can be easily understandable.

The interesting feature of use case is that it combines both the modeling and text-based requirements. A use case diagram provides an overview of an interaction between system and actors to achieve a required goal, while the use case description describes an interaction composed of NL sentences [53]. The success of a use case is dependent on satisfaction of targeted goal.

The use case structure is based on a use case name, Actor, Description, Normal events, Alternative events, Pre-condition and Post-condition.

*Use Case Name:* It states the goal of the use case;

*Actor:* The term actor is used to describe any person or system that interacts with the system to accomplish a goal. A primary actor triggers a system's behavior by initiating an interaction with the system whereas a secondary actor only interacts with the system.

*Description:* The use case description gives a brief overview of what use case is about.

*Precondition:* Its precondition states the situation which must be true before initiating an interaction with the system,
**Post-condition:** The post-condition describes the situation which should be true upon successful completion.

**Normal Events:** The use case normal events lists the expected sequence of interactions to lead to successful achievement of use case goal.

**Alternative Sequence:** The alternative sequence is described in a separate section than the main flow. The alternative sequence consists of scenario other than basic flow. Alternative sequence represents an undesirable path to the user but whenever alternative sequence occurs, the system will ideally react in a way that recovers the flow and provides some useful information to the user.

The quality of use case is discussed as follow:

**Use Case Quality:** The quality of the use case model may largely impact the quality of the resulting software system [25] it is, therefore, essential to develop good quality use cases. Before discussing how to obtain a good quality use case, it is important to understand the quality of the use case.

Nevertheless, there is no commonly agreed theory on how to construct use cases and there are different opinions about what constitutes quality in use case models [25], but there are some findings which defined basic attributes and provided a checklist of use case quality. These include clarity, consistency, correctness, readability, unambiguousness and completeness.

Following are some of the reported guidelines and rules which are used to develop use case.

### 2.6.1 Guiding Use Case Authoring: Results of an empirical study

The CREWS\(^2\) long term research project is a developing software tool working on use cases to validate and elicit system requirements. The CREW team has proposed a set of guidelines to obtain a well-structured and good quality use cases. The CREW model is based on two parts, the Style Guides (SGs) and Content Guidelines (CGs) [95].

\(^2\)The CREWS (Co-operative Requirements Engineering With Scenarios, an EU funded ESPRIT project 21903).
The six SGs defined a set of rules for structuring a use case. SGs prohibited the use of adverb, adjective, synonyms, homonyms, passive voice sentences, sentences with more than two clauses, negative form of sentences and mixing alternatives with main section.

The eight CGs were obtained from the aforementioned theoretical research in linguistics, artificial intelligence and previous applications of Case Grammars to requirements analysis. The CGs suggested rules for contents of a sentence within the use case.

The work in [95] has performed an empirical study in order to examine effectiveness of CREW guidelines. This study included 69 postgraduate students. These students had expertise in object-oriented modeling and were divided into four different groups.

The major results of the study described that the guidelines were supportive to increase the completeness of use case and reduce the chances of making errors while writing use case. It is declared that the guidelines should be simple and clear to obtain a specified goal. The quantitative result also pointed that the guidelines do not work in all cases [95].

2.6.1.1 Analysis of CREW guidelines with 3Cs

- **Clarity**

  *Control to ambiguity:* The use of synonyms, homonyms and negatives sentences can lead to ambiguity, therefore, the SG4 and SG6 can be supportive to obtain clear use cases. The use of pronouns and negative sentences is prohibited in use case according to SG2 and SG4. The FrameNet frames also provide an approach where FEs are used instead of pronouns.

  *Semantics understanding:* There is no explicit support for semantic understanding as given by FameNet frames where each word and FE is annotated with detailed examples. Although SG8 stated that there should be meaningful response of each action, but it did not give any support to semantic understanding of individual word or concept.

- **Completeness**

  The hypothesis related to SG1 and SG3 stated that SG1 is helpful in use case completion as each sentence should be started separately and alternative flow of use case should be written in separate section. The SG1 can be helpful to obtain clear structure of use case
but there is no support to identify missing information as FrameNet frames do. Thus, we can say that it provides partial support to use case completeness.

- **Consistency**

**Interrelated requirements:** Since the SGs and CGs do not provide any support to semantics understandings, therefore it is hard to determine interrelated information. Thus, we can say that guidelines do not provide any support to obtain consistent use case.

**Modularity of concern:** The proposed idea of using SGs and CGs are not supportive in modularity of concern as these guidelines provide just an approach how to structure the flow of use cases. On the other hand the FrameNet frames provide a set of FEs which construct a general human perception about a word.

The experiment of CREW project is replicated by Cox, K. and K. Phalp [96] who rejected the claims of above CREW project.

### 2.6.2 Replicating the CREWS’ Use Case Authoring Guidelines Experiment

Cox, K. and K. Phalp in [96] performed a replication of CREW project discussed above and claimed that the CREW guidelines have no role in improvement of use case description. The author took down the CREWS hypothesis as unclear. They criticized the CREW experiment in three different dimensions.

Firstly, they believed that the CREWS guidelines have failed to influence them. Secondly, the validity of hypothesis was not satisfactory for them. And finally, they were not satisfied with the results that the guidelines could have produced a better set of use cases.

To replicate the CREW guidelines, an experimental study had conducted among fourteen postgraduate students. These students belonged to software engineering background but were not familiar with use cases. The participants were trained with use cases before experiment.

The findings of their results described that although the CREW’s SG gained success in structuring use cases, however, they failed to obtain use case completeness. Similarly, the CG are also failed to obtain complete use cases.
The study had not found any significant difference between control group and experimental group. In fact, the results of experimental group (using both SG and CG) were worst as using both SG and CG together might be too much for participants. It had been proposed that the reduced set CGs could have been equally effective for the study.

2.6.2.1 Analysis of CREW guidelines with 3Cs

- **Clarity**

  *Control to ambiguity:* The SG2 and SG4 have prohibited the use of pronouns and negative sentences writing use case. Therefore, the stated SGs can be helpful to obtain clear use cases.

  *Semantics understanding:* The SGs and CG has no support in semantic understanding.

- **Consistency**

  *Interrelated requirements:* Since SGs and CG has no support in semantic understanding and to identify missing information, thus it does not provide any support to detect interrelated/dependent requirements

  *Modularity of concern:* The author highlighted the shortfalls of CREW guidelines. We have already discussed in section 3.1.1 that the CREW guidelines have no support in modularity of concern.

- **Completeness**

  We have already pointed that the CREW guidelines has no role in obtaining complete use cases and the same point has been claimed by [96] that the SGs were not particularly useful for action completeness or use case completeness.

2.6.3 Comparing Use Case Writing Guidelines

Since CREW guidelines were found lengthy and difficult, therefore, the simplified form of CREW guidelines was introduced by [97]. The CP writing rules has inspired from writing styles of [98]. The CP writing rules consist of Style and Structure rules.
The eight CP style rules are simplified and modified forms of CREW style guidelines. The CP style rules prohibited the use of pronoun, adjective, adverb, negative form of sentences. The style rules further recommended the use of numbered line for each new sentence, present format of verbs and meaningful response to an action, use of consistent agent, object and action names in all action descriptions and logical coherence throughout the description.

The four CP structure rules are compressed and modified form of CREW CG rules. The structure rules suggested the rules to write the content of use case in accurate form. The structure rules suggested that the contents of the use case should follow any of the following structure: “Subject verb object, Subject verb object prepositional phrase, Subject passive” and the name of use case should be underlined.

To assess the quality of the use cases written with CP rules against CREWS guidelines, a pilot study was conducted among twenty four master students of software engineering [97]. These students were divided into four groups. Two groups were asked to develop use cases for two separate systems (ATM and Retail) with CP rules, while the other two groups were asked to develop use cases for ATM and Retail system with CREW Guidelines.

The author described four characteristics of the use case description known as 4Cs which were used to evaluate the quality of use case.

**Coverage**

Completes: does the use case terminate or does the path get caught in a loop?

Rational: does the use case provide a rational answer to the problem?

Span: does the use case contain all that is required to answer the problem?

**Coherence**

Logical order: does the use case follow a logical path? Is this path clear?

Logical coherence [11]: is there coherence through the use case both locally and globally, thus making the use case easier and quicker to read?

Abstraction: is the use case at a consistent level of abstraction throughout? Mixing abstraction levels will cause difficulty in understanding.

**Consistent Structure**
Variations: are alternative paths excluded from the main flow? Inclusion of alternative paths in the main flow reduces readability.

Consistent Grammar: is present tense used throughout? Are adverbs, adjectives, pronouns and negatives avoided?

Sequence: is the numbering of events in the main flow consistent?

**Consideration of Alternatives**

- Separation: is there a separate section for alternative/exceptional paths to the main flow?
- Viable: are the alternatives viable?
- Numbering: do the alternative numberings match the numbers in the main flow?”

The quality of use cases was tested against hypothesis as following:

The CREW guidelines produce better use case than CP rules in terms of efficiency (time taken to write use case and the length) of use case, the number of events that implement rules or guidelines accurately and comprehensibility of the use cases.

The results of the experiment in terms of efficiency have not presented any statistical difference between CP rules and CREW guideline. In terms of comprehension the CP rules were found more suitable than CREW guidelines to have more logically understandable use cases than CREW. As far as guideline rules are concerned, both the styles performed well at the same level. The only difference between CP rules and CREW guidelines is that CP rule are simplified and easy to be used.

2.6.3.1 Analysis of CP Rules with 3Cs

- **Clarity**

  **Control to ambiguity:** The styles of CP rules are helpful to control ambiguity as it constraints the usage of pronoun, adverb, adjective, negative and passive forms of a sentences.

  **Semantics understanding:** Although the CP rules are helpful to write down the clear structure and controlling ambiguity of the use cases, but it does not provide any direct support to semantics understanding of use cases.

- **Completeness**
The author claimed that as it is a simplified form of CREW guidelines, thus it does support the completeness of use case in similar way as CREW guidelines do. However, we already discussed that the CREW guidelines provide guidance of structuring use case that can be helpful in understanding but does not give any explicit support to obtain missing information of use case.

- **Consistency**

*Interrelated requirements:* The CP rules mainly provide support to structure use case. It does not provide any support to determine interrelated requirements.

*Modularity of concern:* The proposed rules provide how to write use case but does not provide any support to modularity of concern.

### 2.6.4 Improving the Quality of Use Case Descriptions: Empirical Assessment of Writing Guidelines

The previous studies have claimed that the CREW guidelines have improved the quality of use case descriptions, but some still considered it more complex and complicated as compared to CP rules. In this regard, Phalp et al [24] performed an empirical work to explore whether CP would possibly result in more envious use case structures and descriptions than given by the CREWS guidelines?

An experimental study took place among 60 participants belonging to software engineering background. Participants were equally divided into four groups. The participants were asked to develop use cases for two different scenarios - ATM and Retail - using two sets of guidelines (CREW guideline and CP rules).

Since CP rules were introduced to provide an easier approach than CREW guidelines in developing use cases, therefore, the hypothesis of the study was set as the CP rules performed better than CREW guidelines while creating use case for same problem. Secondly, the CP rules scored significantly better than the equivalent CREWS where the quality of use case was developed with CREW guideline and CP rules were compared against the use case quality checklist. The use case quality checklist consisted of ‘7Cs of communicability’ (7Cs detail is available in section 2.5.5) [51].

The result of the experiment illustrated that the CP rules were used in greater number than the CREW guidelines for Retail scenario but had no effect on ATM scenario.
Similarly, the use cases developed with CP rules were found better in quality but again these results were reflected in Retail scenario only. The final result described that many users could benefit from the adoption of a minimal (CP) set of rules to improve the quality of use case descriptions.

2.6.4.1 Analysis of CP with CREW guidelines against 3Cs

- Clarity

*Control to ambiguity:* We have already mentioned that a few CREW guidelines and CP rules do not allow the use of negative sentences, pronouns, adverb, adjective and passive form of sentences which could be helpful to control ambiguity.

*Semantics understanding:* There is no direct support in semantics understanding, although the less use of pronouns could help in obtaining clear semantics.

- Completeness

The previous studies about CREW guidelines and CP rules described that well-structured and complete set of use cases could be obtained using instructions.

- Consistency

*Interrelated Information:* The above work proposed a comparison of CP rules and CREW guidelines. It does not provide any support to obtain interrelated information.

*Modularity of concern:* The above work proposed a comparison of CP rules and CREW guidelines. It does not provide any support to modularity of concerns.

2.6.5 Assessing the Quality of Use Case Descriptions

Phalp et al [51] introduced a set of checklist to evaluate the quality of use case. The use case description quality checklists have been derived from theories of text comprehension taken from the discourse processing community. Discourse processing focuses on the ways in which readers and listeners comprehend different languages.

The checklist was mainly derived to improve the communicability of the use case descriptions and introduced 7Cs such as Coverage, Cogent, Coherent, Consistent abstraction, Consistent structure, Consistent Grammar, and Consideration of Alternatives to obtain a well described use case. The vital purpose of 7Cs application is a quick evaluation of a given description enabling the reader to understand use case completely in a shorter duration.
Following are the 7Cs set of heuristics that might be valuable in general validation when writing descriptions.

**Coverage**

*Span:* The use case should contain all that is required to answer the problem.

*Scope:* The use case should only contain the detail relevant to the problem statement. Extra unnecessary information provided is out of problem scope and not required.

**Cogent**

It contains three elements, Text Order, Dependencies and Rational Answer.

*Text Order:* The use case should follow a logical path with events in the description in the correct order.

*Dependencies:* The use case should complete as an end-to-end transaction (which can include alternative / exceptional flows). Does the actor reach a state that stops the transaction from terminating as expected?

*Rational Answer:* The logic of the use case description should provide a plausible answer to the problem.

**Coherent**

The sentence being written should repeat a noun in the last sentence or a previous sentence, if possible. The description is easier to read and quicker to understand if there is logical coherence throughout.

**Consistent Abstract**

The use case should be at a consistent level of abstraction throughout. Mixing abstraction levels (problem domain, interface specification, internal design mixes) may cause difficulty in understanding.

**Consistent Structure**

*Variations:* Alternative paths should be excluded from the main flow. Inclusion of alternative paths in the main flow reduces readability.

*Sequence:* Numbering of events in the main flow should be consistent.

**Consistent Grammar**
The gamer of use case should be consistent, and the present tense should be used. The adverbs, adjectives, pronouns, synonyms and negative should be avoided.

**Consideration of Alternatives**

*Separation:* The alternative section should be a break up section from main section.

*Viable:* Alternatives should be meaningful and make sense.

*Numbering:* Alternative numberings should exactly match the numbers in the main flow.

A case study from industry was selected to illustrate the application of 7Cs. There were total twenty six use cases where only two use cases *Printing Online Customer Account* and *Applying for Trading Account* were tested against 7Cs. A critical feedback was also obtained from stakeholders to provide a validation of the 7Cs in terms of how useful the use cases were to those stakeholders.

The result of the study described three main achievements of using 7Cs for use case. First, this approach can be used to investigate more guidelines for use case. Second, one can test for the desirable quality features in existing descriptions, thus enabling empirical validation. Lastly, that even 7Cs are the quality features but these can be used as a checklist for reviewing use case quality. The result of study described that the guidelines have positive impact on use case quality.

**2.6.5.1 Analysis of 7Cs with 3Cs**

- **Clarity**

  *Control to ambiguity:* The 7Cs can be helpful to control ambiguity to some extent as the 2Cs “Cogent and Consistent Structure” provide suggestion in writing structured use cases and proposed an idea of using consistent grammar. But there is no direct support to deal with synonyms, homonyms as FrameNet do.

  *Semantics understanding:* There is no direct support for semantic understanding, but it has been discussed that 2Cs “Coherent and Consistent Abstract” can be helpful to enhance use case understandability.

- **Completeness**

  The 7Cs provide comprehensive guidance on how to write use cases and evaluate the use cases completeness using 2Cs “Converge and Consideration of Alternatives”.
Therefore, we can say it can be supportive to obtain complete use cases, but it still lacks any support to identify missing information as FrameNet does.

- **Consistency**

**Interrelated Information**: It seems to be that the 7Cs does not provide any support to obtain interrelated information.

**Modularity of concern**: In above work, the author has discussed the importance of mental model in use case description but has not presented any approach that could help in modularity of concern.

### 2.6.6 An Empirical Quality Assessment of Automotive Use Cases

Tormer et al [26] presented a new criteria to evaluate the quality of automotive use. The criteria were based on previous research of [25, 95-97]. The criteria were mainly connected to desired quality attributes: Completeness, consistency, unambiguously, completeness, and readability level of detail. The table 2.3.6-1 below illustrates how the mentioned quality attributes were achieved by CREW guidelines and CP rules.

**Table 2.6.6-1 Evaluation Criteria by[26]**

<table>
<thead>
<tr>
<th>Characteristics</th>
<th>Criteria</th>
<th>Questions</th>
<th>CREWS</th>
<th>CP</th>
</tr>
</thead>
<tbody>
<tr>
<td>C1 Completeness</td>
<td>Missing element</td>
<td>Are one or more elements (Source/Destination/Actor) missing?</td>
<td>CG 1-8 SG5</td>
<td></td>
</tr>
<tr>
<td>C2 Completeness</td>
<td>Goal not achieved</td>
<td>Dose the main flow fail to achieve the goal/purpose?</td>
<td>Coverage: Span</td>
<td></td>
</tr>
<tr>
<td>C3 Correctness</td>
<td>Incorrect flow</td>
<td>Are the wrong steps in the extended flow referenced or is a incorrect return point specified?</td>
<td>Consistent structure: Variations, Consideration of alternatives: Numbering</td>
<td></td>
</tr>
<tr>
<td>ID</td>
<td>Category</td>
<td>Description</td>
<td>Question</td>
<td>Coverage:</td>
</tr>
<tr>
<td>----</td>
<td>------------</td>
<td>------------------------------------</td>
<td>--------------------------------------------------------------------------</td>
<td>----------------------</td>
</tr>
<tr>
<td>C4</td>
<td>Correctness</td>
<td>Outside scope</td>
<td>Are all actions within the scope of the system’s problem domain?</td>
<td>Coverage: Scope</td>
</tr>
<tr>
<td>C5</td>
<td>Consistency</td>
<td>Inconsistent step numbering</td>
<td>Are the steps consistent numbered?</td>
<td>SG2</td>
</tr>
<tr>
<td>C6</td>
<td>Consistency</td>
<td>Irrelevant steps</td>
<td>Is there one step per action? Are all steps/actions relevant for the goal?</td>
<td>Coverage: Scope</td>
</tr>
<tr>
<td>C7</td>
<td>Consistency</td>
<td>Use Case Decomposition</td>
<td>Should the use Case be divided in parts?</td>
<td></td>
</tr>
<tr>
<td>C8</td>
<td>Readability</td>
<td>Misuse of alternative flows</td>
<td>Should alternative flow/include/extend been used?</td>
<td>SG3</td>
</tr>
<tr>
<td>C9</td>
<td>Unambiguity</td>
<td>Unclear alternative flow condition</td>
<td>Is the condition for an alternative flow clearly specified, and at the right location?</td>
<td></td>
</tr>
<tr>
<td>C10</td>
<td>Unambiguity</td>
<td>Incorrect linguistics</td>
<td>Are synonyms, homonyms, pronouns and references unnecessary used?</td>
<td>SG4,SG5</td>
</tr>
<tr>
<td>C11</td>
<td>Level of detail</td>
<td>Will the use case be understandable in 20 years?</td>
<td>Coherence Abstraction</td>
<td></td>
</tr>
<tr>
<td>C12</td>
<td>Misuse of preconditions</td>
<td>Are preconditions</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
To detect defects in use cases against the set criteria, forty three use cases were selected from Volvo Car Corporation (VCC). Each of the use case was evaluated individually by three different reviewers. All three reviewers were from software engineering and electrical engineering background while one had previous experience of use cases at VCC as well.

After use case evaluation, the inspection meetings took place among reviewers where each of the use case was discussed along with the use case defects and the defects were then compiled into defect log. Each of the defects was noted with its type in document. Defect type corresponded to the criteria from which the defects were derived.

The result of the study described that most of the defects related to steps of use case has high defect density due to missing elements and linguistics errors. The defects related to use case has high defect density due to misuse of precondition.

2.6.6.1 Analysis of quality criteria C12 with 3Cs

- Clarity

Semantics understanding: The above set of criteria seems to be supportive to control ambiguity and enhance understandability, but it does not provide any direct support to understand the semantics of text or word.
Control to ambiguity: C9 and C10 seems to be that supportive to detect ambiguities but it has not suggested any approach which can resolve ambiguity as. Thus, we can say that the proposed criteria provide half way support to control ambiguity.

- **Completeness**

The proposed idea does provide support to evaluate completeness in terms of missing elements and goal achievement, but it does not provide any support to obtain relevant information. Thus, we can say that the proposed criteria provide assessment of completeness not completeness by itself.

- **Consistency**

Interrelated requirements: It does not provide any support to determine interrelated requirements.

Modularity of concern: The proposed set of criteria has suggested that to obtain consistence use case, the use case must have consistent number of steps, division of use case in parts and relevancy of steps to the goal. It does not provide any support to modularity of concern.

### 2.6.7 Quality and Understandability of Use Case Models.

Anda et al [25] performed explorative study where use cases were developed by using three different sets of guidelines given as below:

**Minor Guideline:** The Minor guideline provides support in identification of actors and use cases. The actors are identified by transforming information to and from a system. The use cases are identified by looking at main tasks of each actor.

**Template Guideline:** The Template guidelines are those guidelines which offer support in structuring use cases by providing predefined pattern. These guidelines provided a pattern where each use case must have Title, Actors, Trigger, Scope, Summary, Precondition, Basic Flow of Events, Extension Points, Alternate Courses and Post condition.

**Style Guideline:** The results of previous study [95, 96] already described the positive and negative influence of Style guidelines on use cases development procedures. The author has selected Style guidelines as the third set of guidelines in their experiment
from previous recommendation [95] with a slight modifications in the results achieved in [96].

According to this experiment, some original guidelines were considered challenging as they were implemented by a few numbers of participants, whereas some of the guidelines were considered as superfluous as they were used by a large number of participants including those who did not receive these guidelines. The style guidelines suggested that each use case must have a proper flow of events where the flow of events comprised of a number of actions, and the description of each use case must satisfy the guidelines below.

“SG1: Write the UC normal course as a list of discrete actions in the form:

<action#><action description>. Each action description should start on a new line.

Since each action is atomic, avoid sentences with more than two clauses.

SG2: Use the sequential ordering of action descriptions to indicate strict sequence between actions. Variations should be written in a separate section.

SG3: Iterations and concurrent actions can be expressed in the same section of the UC, whereas alternative actions should be written in a different section.

SG4: Be consistent in your use of terminology, that is, use consistent names on actors, objects and actions in all action descriptions. Avoid use of synonyms and homonyms.

SG5: Use present tense and active voice when describing actions.

SG6: Avoid use of negations, adverbs and modal verbs in the description of an action.

Guidelines for content:

CG1:<agent><action><agent>

CG2:<agent><action><object><prepositional phrase>

CG3: ‘If’ <alternative assumption> ‘then’ <list of action descriptions>

CG4: ‘Repeat until’ <repetition condition><list of action descriptions>

CG5:<action 1> ‘while’ <action 2>”

To determine the impact of a proposed set of guidelines. The experimental study was performed among 139 undergraduate students from software engineering background but none of them has any experience of using any guidelines previously. The
participants were given training of use case modeling. The details of guidelines and how to use the guidelines were also taught.

The participants were divided into 31 groups and organized in pairs, where one group was performing a role of developer team and other as a customer. The groups (performing role of developing team) were asked to develop a use case from a given informal requirement specification using any one out of the three sets of guidelines.

Once the use case development was completed, the participants were individually asked to answer a questionnaire. The questionnaire comprised of questions about functionality in the use case, such as how useful the guidelines were, the amount of time spent on both formal and informal specifications, the amount of time spent on communication with their development team and the amount of time spent on reading through the use case model from the development team.

The authors have evaluated the use case models against a set of quality attributes, such as understandability, complete structure, use of consistent, correct and unambiguous terminology.

The result of the study described that the use cases developed using Template guidelines are easier and clear to understand. The Template guidelines provided better understandability and obtained overall the highest score on the quality attributes.

The Minor guidelines were easy to understand and more supportive than Style guidelines in identifying the correct actors and use cases. The use case models developed using the Style guidelines were found the most consistent, for example, the terminologies were used consistently as well as they performed well on the level of detail and realism.

Our results also show that combining the template guidelines with another set of guidelines suitable for documentation of the flow of events of each use case could be more useful.

2.6.7.1 Analysis of 3 sets guidelines with 3Cs

- Clarity

*Control of ambiguity:* The Style guidelines recommended to use consistent terminology and to avoid the use of synonyms and homonyms this is helpful to control
ambiguity but there is no support to identify ambiguities as the FrameNet LUs do by explaining synonyms and homonyms of a word in detail.

**Semantics understanding:** The above guidelines do not provide any support in semantic understanding. However, the Minor guidelines and Template guidelines make the readability of use case easier, but it does not provide any support in semantic understanding.

- **Completeness**

  As described above, the set of guidelines measures the completeness of use case which can be supportive to obtain complete use cases. Although the set of guidelines provide support in identifying use cases, actors, events and flow of events, all lead to a better set of use cases but still, we can see that there is no support to identify missing information.

- **Consistency**

  **Interrelated requirements:** The above guideline does not provide any support to determine interrelated requirements.

  **Modularity of concern:** The discussed set of guidelines has provided no support in modularity of concern.

### 2.6.8 Application of Linguistic Techniques for Use Case Analysis

According to A. Fantechi et al [53], a use case describes the goal and scenario in NL textual description, therefore, it is necessary to identify the defects of NL by analyzing use cases. In this regard, the tools and techniques previously used for textual requirements are utilized to detect defects and obtain a suite of metrics for use case analysis. The major problems associated with a linguistic aspect of use case were grouped into three categories.

**Expressiveness category:** This category deals with understandability of a use case. Ambiguity mitigation and understandability improvement are the part of expressiveness.

Ambiguity mitigation: Ambiguity mitigation helps in detection and correction of ambiguities in the sentences
Understandability improvement: It assesses the level of understandability of requirements specification documents and detect those parts which need to be improved in terms of understandability.

**Consistency category:** This category deals with the presence of semantics contradictions and structural incongruities in the NL requirements document.

**Completeness category:** The characteristics in this category deal with the lack of essential parts within the requirements specifications.

The tools and techniques previously used for textual requirements are used as following;

**QuARS:** QuARS (Quality Analyser for Requirements Specifications) model is composed of high-level quality properties (testability, consistency and understandability) for NL requirements evaluation by means of syntactic and structural indicators. The quality model (see Table 2.5.8-1) is obtained from existing literature [99]. The text document is received by QuARS as an input to detect defective sentences according to quality model where users are free to decide whether to correct the defective sentence or not.

<table>
<thead>
<tr>
<th>Property</th>
<th>Indicator</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Testability</td>
<td>Vagueness</td>
<td>It is pointed out when parts of the sentence hold inherent vagueness, i.e. words having a nonuniquely quantifiable meaning.</td>
</tr>
<tr>
<td></td>
<td>Subjectivity</td>
<td>It is pointed out if sentence refers to personal opinions or feeling</td>
</tr>
<tr>
<td></td>
<td>Optionality</td>
<td>It reveals a requirement sentence containing an optional part (i.e. a part that can or cannot considered)</td>
</tr>
<tr>
<td></td>
<td>Weakness</td>
<td>It is pointed out in a sentence when it contains a weak main verb</td>
</tr>
<tr>
<td></td>
<td>Under-</td>
<td>It is pointed out in a sentence when the subject of the sentence contains a word identifying a class of objects without a modifier specifying an instance of this class</td>
</tr>
<tr>
<td></td>
<td>specification</td>
<td></td>
</tr>
<tr>
<td>Consistency</td>
<td>Under</td>
<td>It is pointed out in a Requirements Specifications Document (RSD) when a sentence contains explicit references to: not numbered</td>
</tr>
<tr>
<td></td>
<td>reference</td>
<td></td>
</tr>
</tbody>
</table>
sentences, documents not referenced into and entities not defined nor
described into the RSD itself

<table>
<thead>
<tr>
<th>Understandability</th>
<th>Multiplicity</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>It is pointed out in a sentence that if the sentence has more than one main verb or more than one direct or indirect complement that specifies its subject</td>
</tr>
</tbody>
</table>

| Implicitly        | It is pointed out in a sentence when the subject is generic rather than specific. |

| Comment Frequency | It is the value of the CFI (Comment Frequency Index). [CFI= NC / NR where NC is the total number of Requirements having one or more comments and NR is the number of Requirements of the RSD] |

| Un-explanation    | It is pointed out in a RSD that when a sentence contains acronyms not explicitly and completely explained within the RSD itself |

**ARM**: The Automated Requirement Measurement Tool (ARM) model is similar to the QUARS model defining some quality indicators for requirements documents. The ARM provides a measure to evaluate the quality of requirements documents. In ARM, indicator is grouped according to its inductive characteristics. Indicators are identified on the basis of words, phrases, and linguistic structures. The text document is received as input by ARM where each line of text is searched for the quality indicator.

**Table 2.6.8-2 Standard ARM indicators [53]**

<table>
<thead>
<tr>
<th>INDICATORS</th>
<th>Imperative</th>
<th>Continuance</th>
<th>Directive</th>
<th>Option</th>
<th>Weak Phrases</th>
<th>Incompletes</th>
</tr>
</thead>
<tbody>
<tr>
<td>Shall</td>
<td>below:</td>
<td>e.g.</td>
<td>Can</td>
<td>Adequate</td>
<td>TBD</td>
<td></td>
</tr>
<tr>
<td>Must</td>
<td>as follows:</td>
<td>i.e.</td>
<td>May</td>
<td>as appropriate</td>
<td>TBS</td>
<td></td>
</tr>
</tbody>
</table>
SyTwo: is a tool developed as a Web application performing the linguistic analysis of an English text. This tool is used to check the ambiguous sentences in requirement document and decide whether or not the text is suitable to be used in a requirement document. This verification is carried out by performing a lexical and syntactical analysis of a given text.

SyTwo partially adopts the QuARS quality model and analyses the English text to detect any defect. The SyTwo constructs the derivation trees for each sentence using NL grammar and detects the ambiguous sentence. The sentences having more than one derivation tree are considered syntactically ambiguous.

SyTwo captures the syntactical structure of a sentence by using knowledge base. A component of SyTwo Cmap illustrates the relationship between different parts of a sentence such as Subject, verb and object.
The QuARS, ARM and SyTwo also provide quantitative evaluation of requirements document specifically in category of Expressiveness. The obtained metrics are as follows:

**Table 2.6.8-3 Metrics suit derived from QuARS, ARM and SyTwo [53]**

<table>
<thead>
<tr>
<th>Metrics</th>
<th>Category</th>
<th>Formula</th>
<th>Rational</th>
</tr>
</thead>
<tbody>
<tr>
<td>Coleman-Liau Formula</td>
<td>Readability</td>
<td>5.89 * (letters / words) - 0.3 * (sentences / (words / 100)) - 15.8</td>
<td>It measures the difficulty in reading the document</td>
</tr>
<tr>
<td>Average number of words per Sentence</td>
<td>Readability</td>
<td>Nw / Ns where Nw= n. of words; Ns = n. of requirement sentences</td>
<td>Short sentences make the requirements document more readable/ understandable</td>
</tr>
<tr>
<td>Continuance Index</td>
<td>Traceability</td>
<td>Ncon/Nreq, where Ncon= n. of continuances in sentences; Nreq = n. of requirement sentences. Continuances are phrases as “the following:” that follow an imperative verb and precede the definition of lower level requirement specification.</td>
<td>The use of continuances indicates well a structured document, but too many continuances indicate multiple, complex requirements</td>
</tr>
<tr>
<td>Comment Frequency</td>
<td>Understandability</td>
<td>Nc / Ns where Nc= n. of comment sentences; Ns = n. of requirement sentences</td>
<td>The comments within the requirements document reduce the risk of misinterpretations</td>
</tr>
<tr>
<td>Directives Frequency</td>
<td>Understandability</td>
<td>Nd / Ns where Nd= n. of directives; Ns = n. of requirement sentences</td>
<td>Directives (i.e. words or phrases that indicate examples or other illustrative information) make the document more understandable.</td>
</tr>
<tr>
<td>Property</td>
<td>Ambiguity</td>
<td>Specification Completion</td>
<td></td>
</tr>
<tr>
<td>-------------------</td>
<td>-----------</td>
<td>--------------------------</td>
<td></td>
</tr>
<tr>
<td><strong>Multiplicity</strong></td>
<td>Understandability</td>
<td>Nmul / Nreq where Nmul = n. of sentences having more than one main verb or more than one direct or indirect complement that specifies its subject; Nreq = n. of requirement sentences.</td>
<td>The presence of multiple sentences makes the requirements document more difficult to be read and understood.</td>
</tr>
<tr>
<td><strong>Vagueness</strong></td>
<td>Ambiguity</td>
<td>NVag / Nreq where NVag = n. of sentences including words holding inherent vagueness, i.e. words having a non-uniquely quantifiable meaning; Nreq = n. of requirement sentences.</td>
<td>The presence of vague sentences increases the level of ambiguity of the requirements document.</td>
</tr>
<tr>
<td><strong>Subjectivity</strong></td>
<td>Ambiguity</td>
<td>Nsub / Nreq where Nsub = n. of sentences refers to personal opinions or feeling; Nreq = n. of requirement sentences.</td>
<td>The presence of subjective sentences increases the level of ambiguity of the requirements document.</td>
</tr>
<tr>
<td><strong>Optionality</strong></td>
<td>Ambiguity</td>
<td>Nopt / Nreq where Nopt = n. of sentences containing an optional part (i.e. a part that can or cannot be considered); Nreq = n. of requirement sentences.</td>
<td>The presence of optional sentences increases the level of ambiguity of the requirements document.</td>
</tr>
<tr>
<td><strong>Weakness</strong></td>
<td>Ambiguity</td>
<td>Nwea / Nreq where Nwea = n. of sentences containing contains a weak main verb; Nreq = n. of requirement sentences.</td>
<td>The presence of weak sentences increases the level of ambiguity of the requirements document.</td>
</tr>
<tr>
<td><strong>Under-specification</strong></td>
<td>Specification Completion</td>
<td>Nusp / Nreq where Nusp = n. of sentences having the subject containing a word identifying a class of objects without a specifier of this class; Nreq = n. of requirement sentences.</td>
<td>The presence of underspecification makes the requirements document not fully specified.</td>
</tr>
<tr>
<td><strong>Implicity</strong></td>
<td>Understandability</td>
<td>Nimp / Nreq where Nimp = n. of sentences having the subject generic rather than specific; Nreq = n. of requirement sentences.</td>
<td>The presence of implicit sentences makes the requirements document prone to be misunderstood.</td>
</tr>
</tbody>
</table>
As a case study, the mobile phone functional requirements document was analyzed using QuARS, ARM and SyTwo tools. The document consisted of one hundred use cases. The metrics value is described in Table 2.6.8-4.

Table 2.6.8-4. The document consisted about one hundred use cases [53]

<table>
<thead>
<tr>
<th>Metrics name</th>
<th>Reference values</th>
<th>Actual Value</th>
<th>Used tool</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 Vagueness</td>
<td>The more it is close to 0 the more unambiguous the requirements document is</td>
<td>4</td>
<td>QuARS/ SyTwo / ARM2.1</td>
</tr>
<tr>
<td>2 Subjectivity</td>
<td>The more it is close to 0 the more unambiguous the requirements document is</td>
<td>0</td>
<td>QuARS/ SyTwo</td>
</tr>
<tr>
<td>3 Optionality</td>
<td>The more it is close to 0 the more unambiguous the requirements document is</td>
<td>0</td>
<td>QuARS</td>
</tr>
<tr>
<td>4 Weakness</td>
<td>The more it is close to 0 the more unambiguous the requirements document is</td>
<td>0</td>
<td>QuARS/ SyTwo/ ARM2.1</td>
</tr>
<tr>
<td>5 Under-specification</td>
<td>The more it is close to 0 the more well specified the requirements document is</td>
<td>19</td>
<td>QuARS</td>
</tr>
<tr>
<td>6 Under-reference</td>
<td>The more it is close to 0 the more consistent the requirements document is</td>
<td>0</td>
<td>QuARS</td>
</tr>
<tr>
<td>7 Implicity</td>
<td>The more it is close to 0 the more understandable the requirements document is</td>
<td>12</td>
<td>QuARS</td>
</tr>
<tr>
<td>8 Unexplaination</td>
<td>The more it is close to 0 the more understandable the requirements document is</td>
<td>0</td>
<td>QuARS</td>
</tr>
<tr>
<td>9 Coleman-Liau Formula</td>
<td>Ranged from 0.4 (easy) to 16.3 (difficult)</td>
<td>17.6</td>
<td>SyTwo</td>
</tr>
<tr>
<td>10 Average number of</td>
<td></td>
<td>14.82</td>
<td>QUARS</td>
</tr>
<tr>
<td>words per sentence</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>11 Continuance Index</td>
<td>0.1 – 0.2</td>
<td>0</td>
<td>ARM 2.1</td>
</tr>
<tr>
<td>12 Comment</td>
<td>0.1 – 0.3</td>
<td>0.04</td>
<td>QuARS</td>
</tr>
</tbody>
</table>
The values of metrics 9, 10 and 14 pointed out that the sentences of the document need to be simplified to decrease the risk of being misinterpreted. The values of metrics 12 and 13 pointed out that the document is missing detailed information which can improve understanding. The values of the metrics 1, 5 and 7 pointed out that we can say that the terminology selected for requirements documents is not appropriate.

The results of the study described that although the linguistics techniques can be helpful to assess requirements in terms of completeness, but it has provided no support to completely address the aspects related to the correctness and the consistency of requirements.

### 2.6.8.1 Analysis of linguistics techniques with 3Cs

- **Clarity**

  **Control to ambiguity:** The set of metrics obtained from linguistics tools such as (Vagueness, Subjectivity, Optionality and Weakness) are helpful to identify ambiguous and less clear sentence. The derivation trees created by ‘SyTwo’ are mainly dealing with ambiguous sentences by setting a sentence ambiguous if it has more than one derivation trees. Thus, we can say that linguistics tools are helpful to identify ambiguous sentences, but these tools failed to find how to solve this ambiguity.

  **Semantics understanding:** The quality indicator “Directives” is supportive to some extent in understanding the requirements

- **Completeness**

  The abovementioned linguistics tools have provided support to measure the completeness of a requirement which indicated that what part of requirements document needs extra information. Although such measurement can be supportive in
obtaining a complete set of requirements, but these tools do not provide any support in obtaining any missing information.

- **Consistency**

*Interrelated requirements:* The QuARS quality model provide supported to determine referential sentences in SRD but still it lacks to point out interrelated concepts as FEs do.

*Modularity of concern:* The abovementioned linguistics tools have not provided any explicit support to the modularity of concern. Although, a component of ‘SyTwo Cmap’ described relationship between different parts of a sentence such as Subject, Verb and Object, we cannot say that it does provide support to modularity of concern.

### 2.6.9 Empirical Evaluation and Review of a Metrics–Based Approach for Use Case Verification*

Bernárdez, B. and A. Durán, [100] performed an empirical evaluation and review of use cases using metric-based verification heuristics. The study was implemented on 127 use cases selected from 8 requirements documents. The selected use cases were developed by software engineering students using REM tool. The students have basic information of RE, use cases and were able to use REM tool.

The REM tool is a free XML–based requirements management tool developed by one of the authors. The use case developed using REM have triggering event, a precondition, a postcondition, and an ordinary sequence of steps describing interactions leading to a successful end.

The proposed set of metrics by [100] is as follows:

<table>
<thead>
<tr>
<th>Metrics</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>NOS</td>
<td>Number of steps of the use case (NOS=NOAS+NOSS+NOUS)</td>
</tr>
<tr>
<td>NOAS</td>
<td>NOAS Number of actor action steps of the use case</td>
</tr>
<tr>
<td>NOSS</td>
<td>NOSS Number of system action steps of the use case</td>
</tr>
<tr>
<td>Metric</td>
<td>Description</td>
</tr>
<tr>
<td>----------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>NOUS</td>
<td>NOUS Number of use case action steps of the use case <em>(inclusions or extensions)</em></td>
</tr>
<tr>
<td>NOCS</td>
<td>NOCS Number of conditional steps of the use case</td>
</tr>
<tr>
<td>NOE</td>
<td>NOE Number of exceptions of the use case</td>
</tr>
<tr>
<td>NOAS_Rate</td>
<td>NOAS_RATE Rate of actor action steps of the use case <em>(NOAS/NOS)</em></td>
</tr>
<tr>
<td>NOSS-Rate</td>
<td>NOSS_RATE Rate of system action steps of the use case <em>(NOSS/NOS)</em></td>
</tr>
<tr>
<td>NOUS-Rate</td>
<td>NOUS_RATE Rate of use case action steps of the use case <em>(NOUS/NOS)</em></td>
</tr>
<tr>
<td>CC</td>
<td>CC Cyclomatic complexity of the use case <em>(NOCS+NOE+1)</em></td>
</tr>
</tbody>
</table>

**NOS (Number of steps) metric:** The first heuristic is about use case length where the NOS should be between (3, 9). It is assumed that the use case with a very few steps can be considered incomplete, whereas the use case having too many sentences can be considered complex.

NOS is a combination of NOAS *(Number of actor action steps of the use case)*, NOSS *(Number of system action steps of the use case)* and NOUS *(Number of use case action steps of the use case *(inclusions or extensions)*).

**NOAS_Rate (NOAS/NOS):** The actor-to-actor interaction does not cause use case defects, but it significantly increases the NOS without making the use cases defective. Thus, the value of NOAS_Rate should be between [30%, 60%].

**NOSS_Rate (NOSS/NOS):** The value of NOSS is always inverse of the value of NOAS. The use case with higher number of NOSS_Rate is considered faulty, such use cases only describe the internal system actions without considering actor participation. Therefore, the value of NOSS_Rate should be between [40%, 80%].

**NOUS_Rate should be in [0%, 25%]:** The understandability of a use case is also dependent on its relationship with other use cases. The excessive use of use case relationships makes use cases difficult to understand; therefore, the proposed heuristics suggested that the NOUS_Rate should be between 0% to 25%.

**Cyclomatic Complexity (CC) (NOCS+NOE+1):** The use cases having many alternative sequences of steps are considered complex and complicated. The number of alternative
paths of a use case can be defined as the cyclomatic complexity (CC) of a use case in the same sense. The CC of use case can be considered as the number of conditional steps plus the number of exceptions plus one.

Using the above metric the author first performed the manual verification of the requirements documents and then the results were compared with properties listed in [76]. The result of the work described that the defective use cases can be determined using proposed set of metrics.

2.6.9.1 Analysis of Metrics against 3Cs

- **Clarity**

  *Control to ambiguity:* There is no support to control ambiguity. The proposed work just demonstrated the evaluation of use case.

  *Semantics understanding:* Similarly, it does not provide any support for semantics understanding as it just evaluates the use case, thus we can say that it provides no support for clarity in terms of ambiguity or semantic understanding although it does provide support to write a use case with clear structure.

- **Completeness**

  The NOS metric is used to measure the completeness of a use case i.e the number of steps should be not less than 3 and not more than 9. Although a very less number of steps can have impact on use case completeness and can be considered as encouraging step to determine use case completeness but it is not very logical that the use case having more than 9 steps will be defective. Thus, we can say that the given criteria provide little support to obtain a complete use case.

- **Consistency**

  *Interrelated requirements:* There is no support to obtain interrelated requirements.

  *Modularity of concern:* The proposed work provides no support to modularity of concern as it provides only the metrics-based heuristics verification of use cases.

2.6.10 A Pattern Language for Use Cases Specification

Da Silva, A.R., et al[101] proposed a pattern language to improve the quality of use cases specifications by checking consistency, completeness and correctness of use case.
The aim of the study is to keep consistency between use cases model and the domain model when dealing with intensive information systems.

The template of proposed pattern consisted of pattern name, context, problem, solution, examples, consequences, related patterns, and known uses. The rules proposed by pattern languages are (P1) Define Use Case Types, (P2) Keep Use Case Consistent With. The Domain Model, (P3) Define Use Case with Different Scenario and Interaction Block Types, and (P4) Define Use Case With Different Action Types.

In this regard the use case has to be categorized in above mentioned four patterns. The use cases of similar type such as to manage with data the use cases create, view, update, delete or search came under same type. The Domain Entity describes the scope of information systems in a domain such as the purpose of a use case is to access or manage data entities, and so each use case is related to some domain entities.

In pattern3, the scenario of use case along with scenario type such as Main Scenario, Alternative Scenario and Exception Scenario should be defined. The actions of the use case- both actor action and system action- should be also defined under Interaction Block type.

To specify more clearly the goal of use cases, it is recommended to define a use case type for each use case. Use case should be defined in the context of a domain model in a way that during the use case specification, you can continuously inspect and check the consistency between the use case model and the domain model.

2.6.10.1 Analysis of pattern languages with 3Cs

- Clarity

*Control to ambiguity:* The pattern languages are helpful to obtain the consistent structure of use case which might be helpful to determine ambiguities. Thus, we can say that it provides moderate support in clarity.

*Semantics understanding:* Since the given pattern keeps use case consistent with the domain, therefore, we can say that somehow the semantic understanding of the use case is improved with patterns.
• **Completeness**

The clear structure of use cases is helpful to enhance use case understanding but we cannot say that pattern provides any support to retrieve relevant information, thus pattern provide little support to obtain complete use case.

• **Consistency**

*Interrelated requirements:* The given patterns are somehow useful in determining related use cases as it keeps the use case consistent with domain but as given pattern does not provide support in understating semantics requirements, therefore, we can say that a given pattern provides little support in determining interrelated requirements.

*Modularity of concern:*

There is no direct support in modularity of concern but since the use case should be consistent with domain, thus we can say that given pattern language provides little support to modularity of concern.

### 2.6.11 Summary

The overall summery of background chapter is demonstrated in table 2.5.11-1, where different notation described the role of previous work played in obtaining clear, complete and consistent use case. The assessment of roles is categorized in four different categories described as follow: Not Supported N/S; Little support: ⬜; Moderate support: ☐; High support: ●

Previous research has stated work on 3Cs but none of them provided fully support to obtain all 3Cs together. Though linguistics tools [53] can be helpful to control ambiguity but none of them provides an approach like FrameNet where all the relevant concepts
(not only meanings of a word) are explained by individual frames. Secondly, these tools can clarify the customer textual requirements but do not provide any support to determine missing requirements as the FEs do. In FrameNet frames, an absolute idea about a single word is expressed via the FEs and frame to frame relation. CL tools do not provide any support to identify interrelated requirements because they do not offer any detailed information of any particular concept.
### Table 2.6.11-1 Role of pervious work played in obtaining Clear Complete and Consistence use case

<table>
<thead>
<tr>
<th>Literature</th>
<th>Clarity</th>
<th>Completeness</th>
<th>Consistency</th>
</tr>
</thead>
<tbody>
<tr>
<td>Computational Linguistics for Requirements Elicitation [65]</td>
<td>![Clarity]</td>
<td>N/S</td>
<td>![Consistency]</td>
</tr>
<tr>
<td>Conceptual modelling using LIDA [67]</td>
<td>![Clarity]</td>
<td>N/S</td>
<td>N/S</td>
</tr>
<tr>
<td>WordNet In Requirement Engineering [69] [16, 71, 89, 91]</td>
<td>![Clarity]</td>
<td>![Completeness]</td>
<td>![Consistency]</td>
</tr>
<tr>
<td>Checklist Based Inspection Approach [80-82]</td>
<td>![Clarity]</td>
<td>N/S</td>
<td>N/S</td>
</tr>
<tr>
<td>Controlled Natural Language needs [83-85] [86]</td>
<td>![Clarity]</td>
<td>![Completeness]</td>
<td>N/S</td>
</tr>
<tr>
<td>Knowledge-Based Approach [87] [88] [89, 90] [87]</td>
<td>![Clarity]</td>
<td>![Completeness]</td>
<td>![Consistency]</td>
</tr>
<tr>
<td>Heuristics Based Approach [93] [94]</td>
<td>![Clarity]</td>
<td>![Completeness]</td>
<td>N/S</td>
</tr>
<tr>
<td>CREW Guide Lines [95]</td>
<td>![Clarity]</td>
<td>N/S</td>
<td>N/S</td>
</tr>
<tr>
<td>CP Rules [24]</td>
<td>![Clarity]</td>
<td>N/S</td>
<td>N/S</td>
</tr>
<tr>
<td>A set of heuristics 7Cs [51]</td>
<td>![Clarity]</td>
<td>![Completeness]</td>
<td>N/S</td>
</tr>
<tr>
<td>Quality criteria [26]</td>
<td>![Clarity]</td>
<td>![Completeness]</td>
<td>![Consistency]</td>
</tr>
<tr>
<td>Three different sets of guidelines [25]</td>
<td>![Clarity]</td>
<td>![Completeness]</td>
<td>N/S</td>
</tr>
<tr>
<td>Linguistic Techniques for Use Case Analysis [53]</td>
<td>![Clarity]</td>
<td>![Completeness]</td>
<td>![Consistency]</td>
</tr>
<tr>
<td>Metrics based approach [100]</td>
<td>![Clarity]</td>
<td>![Completeness]</td>
<td>N/S</td>
</tr>
<tr>
<td>Pattern [101]</td>
<td>![Clarity]</td>
<td>![Completeness]</td>
<td>![Consistency]</td>
</tr>
</tbody>
</table>
Chapter 3

3 Approaching the Problem of Bridging Conceptual Gap

This chapter presents a methodology to ameliorate the shortcomings of the NL in requirement (use cases) elicitation. Section 3.1 explains the aim of the study. Section 3.2 presents a methodology for utilization of FrameNet frames in RE. The proposed methodology is exemplified in section 3.3. Section 3.4 evaluates the completeness, clarity and consistency of use cases developed with frame. Summary of the chapter is given in section 3.5.

3.1 Aims

The reliance on the NL for use cases elicitation often results in many problems, such as missing information, inconsistencies with other requirements, and ambiguities. This technique aims to demonstrate the role of frames as a knowledge base for use case (requirements) elicitation and obtain good quality use case. The methodology is illustrated in figure 3.1-1 to address a number of challenges in RE using unique components of FrameNet, such as frames, frame-to-frame relation, LUs, FEs and annotated sentences. Furthermore, the proposed methodology is applied in two different domains to examine frame usability in different domains and scenario.

3.2 Methodology

This section includes the process applied for use of frames in the proposed method. The process is composed of a set of steps, which is illustrated in Figure 3.2-1 and discussed below.
Figure 3.2-1 Process: Utilizing Frames in Use Case Specification

The process starts with the typical use case specification. For each use case its name and description are compared manually against the set of LUs using FrameNet search engine. The FrameNet LUs can be verbs (e.g., planting, farm, cultivate), nouns (e.g., agriculture, farming, artist), adjectives (heavy, green, fresh) and adverbs (e.g., quickly), thus it is not required to emphasize on verbs and nouns only, as reported in previous work [69].

The matched LUs are then used to extract their respective frames. For each extracted frame, the FEs, FEs descriptions, relations and examples illustrating relations between FEs are retrieved from the FrameNet repository.

The frame and all its FEs are presented to the requirements analyst to facilitate the use case specification process. The analyst can check his/her use case against the conceptually complete set of definitions and relationships within the frame, thus identifying any missing actors, roles, and relationships as well as the additional
constraints (e.g. environmental setting and temporal sequences) which are implied by the “normal” use of the NL frame. Frame-to-frame relations can be used to facilitate further information acquisition.

Although we currently identify the LUs relevant to a given use case by hand, this step can conceivably be automated, as the identification process will simply follow the pseudocode presented below:

I. Select the name and description of each use case as an input text.
II. Parse the input text against the database of LU stored in the FrameNet repository.
III. If a part of the input text matches entry into the LU repository, retrieve the name of the frame related to that LU.
IV. Repeat steps II and III until no more matching LUs are present in the input text.
V. For each named frame of step III retrieve the frame, FEs, FEs descriptions, relations, and examples illustrating relations between FEs from the FrameNet repository.
VI. Present each retrieved frame and its related information to the requirements analyst for consideration.
VII. Stop when no more named frame remains.

To further illustrate this approach, the Create Plat Profile use case of the Garden Management System is selected, which is presented in Appendix A. The description of Create Plat Profile use case is as following “This use case provides information for cultivating a particular plant”.

Using the title and description of this use case as an input text, we search through the FrameNet LUs to see if any matching units are present. In the current research this task (Frame search) is manually performed, using the FrameNet website [36], the screenshot of which is presented in Figure3.2-2.

Thus, the word “cultivate” in this use case description is observed to have a relevant frame associated with it.
The matched LU for “cultivate” is used to retrieve its respective FrameNet frame. The frame Agriculture, retrieved via “cultivate” as shown in Figure 3.2-3.

**Agriculture**

**Definition:**

In this frame an **environment** is **cultivated**. The location or ground where that is cultivated by the **cultivation** is often expressed.

- [Example 1](source)
- [Example 2](source)
- [Example 3](source)

**Figure 3.2-3 FrameNet frame “Agriculture” redrived through its’ “cultivate” LU**

Additionally, the further details associated with the Agriculture frame are also retrieved from the FrameNet repository, as shown in Figure 3.2-4 below:
Frame-to-frame relations (set of related frames) of the FrameNet is used to check which other frames are related to “Agriculture” frame, helping to identify potentially relevant further details. Thus, the “Food_Gather” is observed as related frame to “Agriculture” frame, which indicates the harvesting process is potentially relevant to the current system.

The requirements analyst can now use the frame information to check his/her use case against the conceptually complete set of definitions and relationships within the frame. The information provide to requirements analyst is intended to help in identifying any missing actors, roles, and relationships, as well as the additional constraints (e.g. environmental setting, temporal sequences) which are implied by the “normal” use of the NL frame. This is illustrated in more detail in the following sub-section.
3.3 Feasibility Study

The feasibility of the proposed methodology is demonstrated through use of two systems from different domains: garden management and health observation. The reason for selecting the two different systems is to check the compatibility and usability of frames in different domains.

The Garden Management System (GMS for short) is used as our first explorative study. The overview of the textual description of GMS is available in Appendix A. GMS is an interactive system which aims to provide a year-long support to gardens in scheduling and managing their gardening tasks. The GMS requirements were provided by the University of Leicester gardening association for an assignment of Requirements Engineering and Professional Practice Unit (Module code: CO1106) taught to undergraduate first year students at the department of Informatics, University of Leicester.

The second utilized example is the Health Watcher System (HWS) whose textual description is available in Appendix A. This is a web-based application to collect and manage complaints related to public health. The full details of the HWS are available online [71], along with all use case descriptions for it. The authors of the HWS document are an unnamed software company and three researchers. To avoid repetition, the detailed working through the HWS example and its results are provided in Appendix A [102]. Below we focus on discussion of the GMS example.

3.3.1 Detailed Illustrative Example of Method

To illustrate the application of the approach discussed above, the example of the GMS is demonstrated as following. The detailed example and results of the HWS are available in Appendix A [102]. Following are the implementation of steps of figure 3.2-1.
**Step 1 - Requirements elicitation**

This step describes the use case elicitation from textual description (summary of GMS) of the system. The summary of GMS (initially produced by the University of Leicester Student Union to meet the needs of their Gardening program) was sent to a practitioner requirement engineer with a request to prepare use cases for the given system. The use cases for GMS are: Create Plant Profile, Create User Profile, Log In, Generate Alerts, Create Schedule, Weather Update, and Online Forum. The select frames are (Table 3.3.1-1).

**Step 2 & 3 - Frames search engine**

The next step is to find relevant frames for the listed use cases described. The description of use cases is used as a starting point to search the relevant frames using FrameNet website [36]. The FrameNet search engine displayed a list of relevant LUs along with corresponding set of frames. By checking the definition and example of each frame the relevant frame is selected. The set of use cases with their matching frames, LUs and FEs are detailed in table 3.3.1-1.

<table>
<thead>
<tr>
<th>Use Case</th>
<th>Frame</th>
<th>Core FEs</th>
<th>None Core FEs</th>
<th>LUs</th>
</tr>
</thead>
<tbody>
<tr>
<td>Create User Profile</td>
<td>Becoming a member:“A New_member becomes a member of a socially-constructed Group”</td>
<td>Group and New-member</td>
<td>manner, means, purpose, place, recording, reciprocation, role circumstances, descriptive and explanation.</td>
<td>enroll, join and signup.</td>
</tr>
<tr>
<td>Log In</td>
<td>Membership: “A Member is semi-permanently part of a socially constructed Group”</td>
<td>member and group</td>
<td>Elements, manner, standing, time and place</td>
<td>Associated, belong, member, membership membership, part</td>
</tr>
<tr>
<td>Exlude Member</td>
<td>Exlude_Member: “A former Member of a Group ceases to be a member as a result of the action of an Authority within the Group.”</td>
<td>Authority, member and group</td>
<td>Depicive, Explanation means, time and place</td>
<td>Exclude, excommunicate, excommunication, expel, expulsion</td>
</tr>
<tr>
<td>Create Plant Profile</td>
<td>Agriculture: “An Agriculturist cultivates Food. The location or Ground where the Food is cultivated by the Agriculturist is often expressed”.</td>
<td>Agriculturist, food and ground</td>
<td>circumstances, duration, manner, instruments, mean, place, time,</td>
<td>farm, farming and cultivate</td>
</tr>
<tr>
<td>Food_Growing</td>
<td>Food_Growing: “A Grower endeavors to tend to and influence the environment of Food so as to foster its growth.”</td>
<td>Food and grower</td>
<td>circumstances, duration, manner, instruments, mean, place, time, Particular_Iterations</td>
<td>Grow and raise</td>
</tr>
<tr>
<td>Food_Gathering</td>
<td>Food_Gathering: “A Gatherer removes Crop ripe and ready to an accepted degree. The Crop comes from a Source where it has been grown and matured”.</td>
<td>Gatherer, crop and source</td>
<td>Circumstances, duration, manner, means, frequency, mean, place, time, circumstances.</td>
<td>Bring, gather, harvest and pick</td>
</tr>
<tr>
<td>Generate Alert</td>
<td>Telling: “A Speaker addresses an Addressee with a Message, which may be indirectly referred to as a Topic.”</td>
<td>addressee, medium, message, speaker and topic</td>
<td>descriptor, iterations, manner, means, place, time, viewpoint</td>
<td>Advise, apprise, assurance, assure, brief, confide, inform, let know, notification, notify, tell</td>
</tr>
<tr>
<td>Category</td>
<td>Description</td>
<td>Keywords</td>
<td>Example</td>
<td></td>
</tr>
<tr>
<td>---------------</td>
<td>-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
<td>--------------------------------------------------------------------------</td>
<td>--------------------------------------------------------------------------</td>
<td></td>
</tr>
<tr>
<td>Warning</td>
<td>“A Speaker warns an Addressee by giving a Message that describes an undesirable situation that may affect the Addressee. The warning may alternatively be described as about Topic pertaining to the undesirable situation, or issued via a specific Medium.”</td>
<td>descriptor, iterations, manner, means, time and place.</td>
<td>Alert, forewarn, warn, warning</td>
<td></td>
</tr>
<tr>
<td>Weather Alter</td>
<td>“A Speaker warns an Addressee by giving a Message that describes an undesirable situation that may affect the Addressee. The warning may alternatively be described as about Topic pertaining to the undesirable situation, or issued via a specific Medium.”</td>
<td>communicator, _force, Descriptor, iterations, Manner, Means, Place, Time</td>
<td>Alert, forewarn, warn, warning</td>
<td></td>
</tr>
<tr>
<td>Create Schedule</td>
<td>Timetable: “A Timetable of Events is a mental object deliberately created, modified or kept by a Cognizer. The Timetable is often modified by an adjective depicting a Period_of_iterations or a dependent noun expressing the Events of the Timetable. Note that this frame does not include the physical representation of a timetable.”</td>
<td>cognizer, events and timetable</td>
<td>Agenda, itinerary, schedule and timetable</td>
<td></td>
</tr>
<tr>
<td>Weather Update</td>
<td>Weather: “Ambient conditions of temperature, precipitation, windiness, and sunniness pertain at a certain Place and Time. Further Specification of the conditions that pertain may also be indicated”.</td>
<td>place and time</td>
<td>Snowstorm, snowy, storm, stormy, sunshine, thunderstorm, weather</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Precipitation: “Water in some solid or liquid form (the Precipitation) falls from the sky at a particular Place and Time, lasting for a particular Duration.</td>
<td>place, precipitation and time</td>
<td>Downpour, drizzle, drizzle,hail, precipitation, rain event, rain, rain,</td>
<td></td>
</tr>
</tbody>
</table>
The Rate or Quantity of precipitation may also be indicated.”

Online Forum Discussion: “Two (or more) people (the Interlocutors, also encodable as Interlocutor_1 and Interlocutor_2) talk to one another. No person is construed as only a speaker or only an addressee. Rather, it is understood that both (or all) participants do some speaking and some listening--the process is understood to be symmetrical or reciprocal. This frame differs from the Chatting frame in that the Interlocutors have a particular question that they are attempting to decide or understand”.

Interlocution descriptive, amount_of discussion, domain, duration, containing_event, language, manner, mean, means_of_communication

Communicate, communication, confer, conference, consultation, debate, debate, dialogue, discuss, discussant, discussion, exchange, interlocutor, meeting, negotiate, negotiation, parley, parley, talk (to), talk over, talk

The example of Create Plant Profile use case is described in this section, while the detailed findings for other six use cases are given in [102].

In Table 3.3.1-2, the UC1 (Use Case 1) is developed by an independent requirements analyst (without frames) which is then expanded upon in UC-F1 (use case with frame) using FrameNet frames. The Agriculture frame is identified as a corresponding frame to the textual “summary” or description of the Create Plant Profile use case, triggered by the “cultivate” LU. Here, the Actors (line Actor) are represented by User and the Gardener respectively, where Gardener takes on the role of the Agriculturist FE. The steps added in UC-F1 from Agriculture, Food_Growing and Food_Gathering frames are denoted by blue, green and brown color respectively.
Table 3.3.1-2 Use case without Frame (UC) VS Use case with frame (UC-F)

<table>
<thead>
<tr>
<th>Use Case ID</th>
<th>UC1</th>
<th>UC-F1</th>
</tr>
</thead>
<tbody>
<tr>
<td>Use Case Name</td>
<td>Create Plant Profile</td>
<td>Create Plant Profile</td>
</tr>
<tr>
<td>Actor</td>
<td>User</td>
<td>Agriculturist (Gardner.Agr)</td>
</tr>
<tr>
<td>Summary</td>
<td>Provide information for cultivating a particular plant.</td>
<td>Provide information for planting a particular plant.</td>
</tr>
<tr>
<td>Pre-condition</td>
<td>User must be logged into system</td>
<td>User must be logged into system</td>
</tr>
<tr>
<td>Post-condition</td>
<td>A new plant profile is created</td>
<td>A new plant profile is created</td>
</tr>
<tr>
<td>Relation</td>
<td>Extend: UC_2 (Alert)</td>
<td>Extend: UC_2 (Alert)</td>
</tr>
<tr>
<td>Normal events</td>
<td>1. The User selects Choose Plant menu</td>
<td>1. The User selects Choose Plant menu</td>
</tr>
<tr>
<td></td>
<td>2. The system displays a list of plants</td>
<td>2. System asks about the place (geographical location) of user where he/she intends to grow the plant (place. Agr)</td>
</tr>
<tr>
<td></td>
<td>3. The User selects the plant (one only) from available list.</td>
<td>3. User provides the information about his/her location. (place. Agr)</td>
</tr>
<tr>
<td></td>
<td>4. System displays <em>more information</em>, <em>confirm plant</em>, and <em>choose plant</em> options to the user</td>
<td>4. System displays the list of plants can grow at that particular place.</td>
</tr>
<tr>
<td></td>
<td>5. User clicks on <em>more information</em> option for this plant.</td>
<td>5. The system displays a list of plants <em>available for the given location</em>.</td>
</tr>
<tr>
<td></td>
<td>6. System opens a new page with the information about plant, stating:</td>
<td>6. The User selects the plant (one only) from available list.</td>
</tr>
<tr>
<td></td>
<td>6.1 Where to grow the plant</td>
<td>7. System displays <em>more information</em>, <em>confirm plant</em>, and <em>choose plant</em> options to the user</td>
</tr>
<tr>
<td></td>
<td>6.2 What time to grow the plant.</td>
<td>8. User clicks on <em>more information</em> option for this plant.</td>
</tr>
<tr>
<td></td>
<td>6.3 Ideal spacing between plants for sowing.</td>
<td>9. System opens a new page with the information about plant, stating:</td>
</tr>
<tr>
<td></td>
<td>6.4 How to dry the herbs (if the selected plant is a herb)</td>
<td>9.1 Where to grow the plant (<em>physical location, e.g., indoor, outdoor; Ground FE in Agriculturist</em>)</td>
</tr>
<tr>
<td>Step</td>
<td>Description</td>
<td></td>
</tr>
<tr>
<td>------</td>
<td>-------------</td>
<td></td>
</tr>
<tr>
<td>8.</td>
<td>The system returns to the more information and confirm plant options page.</td>
<td></td>
</tr>
<tr>
<td>9.</td>
<td>If the user chooses choose plant option, the steps 2 – 9 are repeated.</td>
<td></td>
</tr>
<tr>
<td>10.</td>
<td>The user chooses the confirm plant option to create a plant profile in his/her growing scheme.</td>
<td></td>
</tr>
<tr>
<td>11.</td>
<td>The system creates a new plant profile for the selected plant.</td>
<td></td>
</tr>
<tr>
<td>9.2.</td>
<td>What time to grow the plant (i.e., month to sow; Time FE in Agriculturist, this is dependent on place)</td>
<td></td>
</tr>
<tr>
<td>9.3.</td>
<td>Ideal spacing between plants for sowing.</td>
<td></td>
</tr>
<tr>
<td>9.4.</td>
<td>How to dry the herbs (if the selected plant is a herb)</td>
<td></td>
</tr>
<tr>
<td>9.5.</td>
<td>The tools need to sow/grow the plant. (Instrument FE in Agriculturist)</td>
<td></td>
</tr>
<tr>
<td>9.6.</td>
<td>How often the user can grow this plant (Frequency FE in Agriculturist)</td>
<td></td>
</tr>
<tr>
<td>9.7.</td>
<td>The outcome of the plant (Outcome FE in Agriculturist)</td>
<td></td>
</tr>
<tr>
<td>9.8.</td>
<td>The duration plant/food needs to grow (Duration FE in Food_Growing)</td>
<td></td>
</tr>
<tr>
<td>9.9.</td>
<td>The starting state of a plant i.e. seed/plant. (Source FE in Food_Growing)</td>
<td></td>
</tr>
<tr>
<td>9.10.</td>
<td>Particular-iteration required by a plant to grow (Iteration FE in Food_Growing)</td>
<td></td>
</tr>
<tr>
<td>9.11.</td>
<td>The amount of effort put into the food gathering event (Degree FE in Food_Gathering)</td>
<td></td>
</tr>
<tr>
<td>9.12.</td>
<td>When to pick the food (Time FE in Food_Gathering)</td>
<td></td>
</tr>
<tr>
<td>9.13.</td>
<td>How often the food can be gathered. (Frequency FE in Food_Gathering)</td>
<td></td>
</tr>
<tr>
<td>9.14.</td>
<td>How long the food takes to be gathered (Duration FE in Food_Gathering)</td>
<td></td>
</tr>
<tr>
<td>10.</td>
<td>The user chooses the information page for the plant he/she intends to grow.</td>
<td></td>
</tr>
<tr>
<td>11.</td>
<td>The system returns to the more information and confirm plant options page.</td>
<td></td>
</tr>
<tr>
<td>12.</td>
<td>If the user chooses choose plant option, the steps 4 – 12 are repeated.</td>
<td></td>
</tr>
<tr>
<td>13.</td>
<td>The user chooses the confirm plant option to create a plant profile in his/her growing scheme.</td>
<td></td>
</tr>
<tr>
<td>14.</td>
<td>The system creates a new plant profile for the selected plant.</td>
<td></td>
</tr>
</tbody>
</table>
Step 4- Refinement of use case

This step presents the requirements generated with help of FEs. Table 3.3.1-3 represents the list of textual statements triggered by different FEs of Agriculture, Growing_Food and Food_Gathering frames. The text defined in GMS textual summary is represented by white row, while the missing requirements which are triggered with FEs (not described in system overview) are represented by gray row.

In above example, the Agriculturist FE is linked with actor (garden manager), the FE Time describes the text “when to sow the plant” defined in GMS summary. The GMS summary did not describe some feature required for gardening activity which are elaborated (denoted by grey row) with FEs Duration, Instrument, Circumstances and Place Frequency. The Food_Gathering is subframe of Agriculture frame, which covers the requirements of harvesting as described in textual summary of GMS.
<table>
<thead>
<tr>
<th>Frame</th>
<th>Steps added</th>
<th>Frame Elements</th>
<th>Textual description</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>in use case1</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Agriculture</td>
<td>9.1</td>
<td>Ground</td>
<td>where to sow the plant- indoor, outdoor pitpot</td>
</tr>
<tr>
<td></td>
<td>9.2</td>
<td>Time</td>
<td>when to sow the plant</td>
</tr>
<tr>
<td></td>
<td>9.5</td>
<td>Instrument</td>
<td>The Instrument is used by the Agriculturist as a tool to help obtain Food. (This tool is not limited to typical artifacts and can be hands, arms, and other atypical hunting and fishing instruments.)</td>
</tr>
<tr>
<td></td>
<td>2,3,4,5</td>
<td>Place</td>
<td>The greater geographic location where the Agriculturist cultivates Food.</td>
</tr>
<tr>
<td></td>
<td>9.6</td>
<td>Frequency</td>
<td>How often the Agriculturist cultivates Food.</td>
</tr>
<tr>
<td></td>
<td>9.7</td>
<td>Outcome</td>
<td>The end result of whole process, the failure or success of an attempt.</td>
</tr>
<tr>
<td>Growing_Food</td>
<td>9.8</td>
<td>Food</td>
<td>It clears the concept that food is not always traditionally eaten stuff but can be flowers etc</td>
</tr>
<tr>
<td></td>
<td>9.9</td>
<td>Duration</td>
<td>For what length of time the Agriculturist attempted to obtain Food</td>
</tr>
<tr>
<td></td>
<td>9.7</td>
<td>Source</td>
<td>The starting state of the Food, e.g., a seed.</td>
</tr>
<tr>
<td>Food_Gathering</td>
<td>9.11</td>
<td>Degree</td>
<td>The amount of effort spent by gatherer to gatherer the food/crop</td>
</tr>
<tr>
<td></td>
<td>9.12</td>
<td>Time</td>
<td>when to pick the fruit</td>
</tr>
<tr>
<td></td>
<td>9.13</td>
<td>Frequency</td>
<td>How often the Gatherer collects his Crop.</td>
</tr>
<tr>
<td></td>
<td>9.14</td>
<td>Duration</td>
<td>For what length of time the Gatherer was gathering the Crop.</td>
</tr>
</tbody>
</table>
3.3.2 Clarity, Completeness, and Consistency of FrameNet enhanced use case

In this section, we compare the clarity, completeness and consistency of use cases developed with and without frames to check either the frames or FEs improve the quality of use cases.

3.3.2.1 Clarity

The clarity of use cases is evaluated based on two attributes, namely semantic understanding and ambiguity.

- Semantic understanding
  The frames perform an innovative role to understand the meanings of a text or a word in different context. The description of FEs (as shown in Table 3.3.1-3) helps to clarify how the actors, as well as other FEs are related to each other, and what roles they undertake, leading to a clearer semantics of the notions discussed in the requirements. Here, the actors (line Actor) are represented by user and the gardener respectively where gardener takes on the role of the Agriculturist FE. The Time and Ground FEs have triggered clarification of the notions “when to sow the plant” and “where to sow the plant” which are already reflected in UC1 by steps 6.1 and 6.2 respectively. Thus, the items, roles and actions required to be in gardening process are clarified by frames and FEs.

- Control to Ambiguity
  Controlling ambiguity is another big challenge in requirements elicitation. The frames also play vital role to control ambiguity. In the above example, the Agriculture frame distinguishes the Ground and Place elements. The Ground element refers to the physical location, i.e. “where to sow the plant”, such as indoors or outdoors, while the Place element refers to geographic location of the garden. Since synonyms (Different words having same meanings) and homonyms (words having same spelling or pronunciation but give different meanings) are one of the causes of ambiguity, therefore we can say that the ambiguity is controlled with FEs and LUs of frames.

3.3.2.2 Completeness

The frames also make supporting part in use case completeness. In the above example, the Agriculture frame exposes a number of additional elements: Instrument, Duration,
Place, Circumstance and Outcome which relate to perform “other general maintenance of plant”. The notion of such maintenance is missing from the original use case (UC1) though it is noted in the project brief. Using these FEs, the additional information is provided in UC-F1 as in sub-steps 9.5 to 9.8 of step 9 (marked with blue colour in Table 3.3.1-2). This explains to the gardener that what tools are required for growing the given plant, how often it can be grown within a year, whether it should be planted from a seed or seedling, and what will the plant grow into?

3.3.2.3 Consistency

The confliction among interrelated requirements leads to incontinency. The FEs provide provision for identifying interdependencies among requirements by explaining different concerns with FEs’s example. The FEs Place, Frequency, Purpose, Time, Duration and Outcome are related with other concern and can be helpful to identify the depended requirements. For instance, the FE Outcome is not the direct part of “Create Plant profile” use case, but it is dependent on Discussion frame where gardeners share their experience with each other (stated in GMS overview). Thus, the use case “Online-forum” is dependent on the use case “Create Plant profile”. The information related to use case “Weather Update” is also dependent on geographical location, thus the FE Place of Agriculture frame has identified relationship between the use case “Create Plant profile” and “Weather Update”.

3.4 Summary

In this chapter, the proposed methodology is demonstrated to overcome the problem of conceptual mismatch in NL requirements by understanding semantics of text-based requirements. In this regard, the Berkeley FrameNet frame project is used as linguistic knowledge base for use case elicitation. The frames and FEs are used to enhance the completeness and understanding of use cases. We illustrated an example of GMS that how a FrametNet frame can support use case elicitation activity by providing suggestions through its FEs, how actors should be organized, what roles should be fulfilled for the coherence of a given concept, and what sequence of interactions could be expected for it.
Our preliminary study indicated that FrameNet frames are a valuable prompt for enhancing semantic understanding of text-based requirements, as well as for identification of missing and related requirements.

We intend to conduct a wider experimental study to compare the quality of use cases produced with and without FrameNet support. To measure such quality, we will adopt a set of metrics for assessing the clarity, completeness and consistency of use cases discussed in Chapter 4.
Chapter 4.

4 Towards Completeness, Clarity and Consistency

This chapter presents an experimental study to determine the role of FrameNet frames in obtaining clear, complete and consistent use cases. Section 4.1 describes research question of the study. Section 4.2 outlines the methodology of adopted evolution. Section 4.3 illustrates result and discussion. Section 4.4 discusses threats to validity. Finally, Section 4.5 gives the overall summary of conducted study.

4.1 Research Question

The study in previous chapter indicates that FrameNet can produce good quality use cases. In this chapter we examine whether the FrameNet provides support to obtain clear, complete and consistent use cases. To this end, we conducted qualitative studies involving students working on use case elicitation. The research question and hypothesis were:

*RQ1: Does fframenet help to obtain Clear, Complete and Consistence use cases?*

*H1: FrameNet is helpful to obtain Clear, Complete and Consistent use cases.*

4.2 Methodology

This section describes the overall process of experiment design to evaluate the quality of use cases developed with and without frames. The first step of the study was to recruit appropriate subjects for the study and train them for the experiment. The suite of metrics, by Torner et al [26], are adopted to analyse use cases quality in terms of completeness, consistency and clarity. The study aimed to discover whether there was a difference between the quality of use cases developed by experiment group and the control group. The above study is a semi-controlled experiment as we had no control to determine how students are developing use cases and what sequence they are following by writing use cases.
4.2.1 Sampling

Sampling is a well-established scientific technique and a fundamental principle of many research methodologies to establish the likelihood or probability. Considering time constraint, the participants of this study were selected based on their availability. Therefore, convenience sampling [103] was used to recruit subjects for the study. In convenience sampling, the selection of subjects is dependent on their convenient accessibility and proximity to the researcher.

Since this study was designed to explore the difference in the quality of use cases produced with and without reference to frames, participants from a software engineering background were selected. The participants were 2nd year BSc in Software Engineering students at the Institute of Management Sciences (IMS) in Peshawar, Pakistan. These students had previously studied a Software Engineering module and were well familiar with the use cases as a requirements elicitation technique. The participants were contacted by one of their tutors who had taught them Software Engineering course previously. Since convenience sampling was used to recruit the participants, students with availability and voluntary willingness to participate in the experiment were recruited. The participants were given consent form for their agreement to take part in the study.

The total number of participants selected for this study was ten. These participants were further divided into two groups the control group and an intervention group. Both groups consisted of equal participants (five in each group).

4.2.2 Training and Material

We delivered a brief (15 minutes long) recap lecture on use case construction to all participating students. The students were divided into two groups. The control group was asked to return after a break, while the second the intervention group was given another 15 minutes lecture on Frames and use of FEs to verify that the relevant element/interaction is represented within the use case.

Thereafter, all students were asked to develop three use cases for the given textual description of Restaurant Management System. They were also provided with the most commonly used template suggested by [21] (shown in Table 4.2.2-1) and asked to develop use cases according to the given template. The participants were given three
use cases, namely "Input Order", "Print Bill" and "Input Payment Details", as shown in Table 4.2.1. The names, goals, and descriptions of use cases were already stated and the participants had to identify actors, precondition, postcondition and normal events of use cases. The material of training is available in Appendix B.

<table>
<thead>
<tr>
<th>Table 4.2.1 Selected Use Cases for experiment</th>
</tr>
</thead>
<tbody>
<tr>
<td>Input Order</td>
</tr>
<tr>
<td>Actors?</td>
</tr>
<tr>
<td>G: To input an order for a Meal</td>
</tr>
<tr>
<td>Description: When a Diner orders a meal, the Waiter writes down the order and puts it into the system. The system presents this order to the Kitchen staff who prepare the food</td>
</tr>
</tbody>
</table>

The intervention group also received a printout of the three frames - one selected for each of the use cases. These are Request Entity frame for Input Order use case, Billing frame for Print Bill use case, and Commerce Pay frame for Input Payment Details use case (details in Appendix B). All participants were asked to write down the basic flow (normal events) of these use cases, list the actors for each use case, precondition and post-condition. The steps in data collection are as follow:
With FrameNet

i. Consent form and information sheet

ii. Pre-study lecture on use cases for 15 minutes (including question & answer).

iii. Further 15 minutes lecture on introduction of FrameNet and its usage in use cases.

iv. Handed over printout of given outline case study of Restaurant Management.

v. Handed over printout of selected use cases and selected frames (one frame selected for each of the use cases).

vi. Participants were asked to create use cases using frame and system description of selected case study.

vii. Participants were given 45 minutes to complete task.

viii. Participants perform their task voluntarily.

Without FrameNet

i. Consent form and information sheet

ii. Pre-study lecture on use cases for 15 minutes (including question & answer).

iii. Participants were given 15 minutes break.

iv. Handed over printout of given outline case study of Restaurant Management.

v. Handed over printout of selected use cases.

vi. Participants were asked to create use cases using system description of selected case study.

vii. Participants were given 45 minutes to complete task.

viii. Participants perform their task voluntarily.

The study was conducted under exam condition where the participants were not allowed to share any information with each other. A light refreshment was served at the end of the study.

4.2.3 Data Analysis

In accordance with the set task, the use cases developed with and without Frames by participants are checked against criteria of completeness, consistency and clarity. To evaluate the completeness, consistency and clarity of use cases, we have used a (subset
of ) metrics proposed by Torner et al [26]. The detail of each metrics is described as follows

4.2.3.1 Completeness

Here, completeness is measured in terms of missing actors of the use case and achievement of the use case goal. The measurement of missing actors is presented with Numeric value while the measurement of goal achievement is presented with True/False values.

(i) Are there any missing actors? and (ii) If the goal of the use case is achieved by the main flow of the provided use cases.

(i) **Missing actors:** An actor is considered missing if the body of the use case discusses an interaction with an actor, but the “Actors” section of the use case does not list that actor as relevant to the use case. The criteria to find missing actors is as follow:

Measurement of Missing actors: (No. of actors in actors section) - (No. of actors discussed in main flow)

(ii) **Goal Achieved:** The criteria for goal achieved is simply dependent achievement and failure of goal outlined in table 4.3.1. This measurement returns true or false value.

4.2.3.2 Consistency

Consistency is measured in terms of freedom from irrelevant steps and semantic conflicts within the use cases.

(i) **Irrelevant steps:** The steps are considered irrelevant if the precondition, post condition and alternative events are mixed with main flow of use case. The information out of the use case scope is also considered irrelevant. The detail example is discussed in section 4.3.2.

Measurement of Irrelevant steps: (No. of main steps of use case) / (No. of Fully Irrelevant steps)

(ii) **Semantic conflict:** The ambiguous and incompatible requirements lead to semantics conflict and disturb the execution of other use case. To overcome the problem of inconsistency it is required to understand the semantics of relationships between requirements [30].
Measurement of Semantic conflicted steps: (No. of main steps of use case) / (No. steps cause of Semantic confliction)

4.2.3.3 Clarity

Clarity is measured in terms of readability and unambiguous structure of main/alternative flows. In this study, we are mainly emphasising on unambiguous structure of main/alternative flows and skipped the linguistics-based readability metrics as the results of readability are overly affected by the English language skills of the study participants.

Measurement of clarity = (No. of main steps of use case) / (No. Fully mixed steps of use case).

4.3 Results & Discussion

This section describes the overall results obtained from conducted study. Here the completeness, consistency and clarity of each use cases are measured using set of metrics (discussed above). The average values of use cases with and without are compared to determine the effect of frames on quality of use cases. Results are presented as follow:

4.3.1 Completeness

Table 4.3.1-1 illustrates the number of missing actors for each use case which is represented by a numeric value. Whereas, for goal achievement the value is represented by true/false values where goal achieved, and goal failed is presented by A and F respectively. In the control group, 2 out of 5 specifications fail to record the customer’s order with the system, 3 out of 5 specifications fail to print the bill (Print Bill use case), and 2 out of 5 specifications fail to record payment details (Input Payment Details use case).

The example of failures in use cases without frame is as follows: The two failures of the Input Order use case in the control group do not discuss inputting order but interact with the system for checking dish availability or bypass using the software system altogether. The two failures of the Print bill use case to achieve the goal were because of that the specification does not talk about printing a bill and has mentioned nothing about actual printing process. Similarly, the three failures of Input Payment details use
case in achieving its goal is caused by complete absence of steps related to inputting payment details.

In terms of missing actors (see numeric values of Table 4.3.1-1) for (Input Order Detail use case) all the 5 use cases have missed 1 or more actors with average of 1.4. Similarly, for (Print Bill use case) 4 out of 5 use cases have missed 1 or more actors with average of 1.2 and for (Input Payment Details use case) 3 out of 5 use cases have missed 1 or more actors with average of 0.8.

Table 4.3.1-1 Completeness: Actors Missed - Goals Achieved/Failed for use case

<table>
<thead>
<tr>
<th>Use Case</th>
<th>Input order Details</th>
<th>Print Bill</th>
<th>Input Payment Details</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>With Frame</td>
<td>Without Frame</td>
<td>With Frame</td>
</tr>
<tr>
<td>UC1</td>
<td>0, A</td>
<td>1, F</td>
<td>1, A</td>
</tr>
<tr>
<td>UC2</td>
<td>0, A</td>
<td>2, A</td>
<td>0, A</td>
</tr>
<tr>
<td>UC3</td>
<td>2, A</td>
<td>1, A</td>
<td>0, F</td>
</tr>
<tr>
<td>UC4</td>
<td>0, A</td>
<td>2, A</td>
<td>0, A</td>
</tr>
<tr>
<td>UC5</td>
<td>1, A</td>
<td>1, F</td>
<td>0, F</td>
</tr>
<tr>
<td>Average</td>
<td>0.6, F0/5</td>
<td>1.4, F 2/5</td>
<td>0.2, F 2/5</td>
</tr>
</tbody>
</table>

In intervention group table 4.3.1-1 shows that the failures to achieve the goals of Input Order, Print Bill and Input Payment Details are respectively 0 out of 5, 2 out of 5 and 0 out of 5. The two failures of Print Bill use case are caused by respectively “generating” the bill and “bringing” the bill without explicitly noting that it has been printed after generation or before bringing it to the customer. This case is similar to one of the control group failures.

Similarly, in terms of missing actors for (the Input Order Detail use case) 2 out of 5 use cases have missed 1 or more actors with average 0.6. Similarly, for (Print Bill use case) 1 out of 5 use cases has missed 1 actor with average of 0.2 and for (Input Payment Details use case) 1 out of 5 use cases have missed 1 actor with average of 0.2.

86
For the Input Order use case in the control group there were twice as many actors missing as in the intervention group. Similarly, for the Print Bill and Input Payment Detail cases, the difference is even more significant, there were respectively 6 and 4 times more actors missed in control group as compared to intervention group.

The average of missing actors for use cases Input Order, Print Bill and Input Payment Details is 1.4, 1.2 and 0.8 respectively. On other hand, average of missing actors for use cases Input Order, Print Bill and Input Payment Details is 0.6, 0.2 and 0.2 respectively. This is not surprising, as the frame explicitly draws the attention to the actors and interactions within it. For example, the definition of Request Entity frame describes that “A Customer requests an Entity from a Supplier. The Customer can order the Entity himself or via a Medium”. So, it shows an interaction of an actor Customer (FE representing role of dinner) with another Actor Supplier (FE representing role of Waiter/Staff) for Entity (FE representing food).

Thus, with respect to completeness, the intervention group performs substantially better both in terms of actor identification and goal achievement.

### 4.3.2 Consistency

Consistency is related to absence of conflicts and misunderstandings across use cases. we have not identified any semantic conflicts within the dataset collected for this preliminary evaluation, we have identified a number of irrelevant steps, i.e., steps that do not at all contribute to the achievement of the use case goal.

The steps of the both groups are summarized table 4.3.2-1 in by illustrating the percentage of irrelevant steps out of the total number of steps defined in each use case. The irrelevant steps in use cases is mostly caused by misunderstanding the goal of use case. For instance, a step stating that “the kitchen staff makes food” (by one of participant without Frame) has no role in the Input Order detail use case, because the requirement related to prepare food should be part of Order Preparation use case. Similarly, a step in the Print Bill use case stating that “For printing a bill the waiter will give all the payment details. Since the payment details (by cash or card) have no bearing on the bill, this step is irrelevant.
Table 4.3.2-1 Consistency: Percentage of Irrelevant Steps

<table>
<thead>
<tr>
<th>Use Case</th>
<th>Input order Details</th>
<th>Print Bill</th>
<th>Input Payment Details</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>With Frame</td>
<td>Without Frame</td>
<td>With Frame</td>
</tr>
<tr>
<td>UC1</td>
<td>73%</td>
<td>25%</td>
<td>25%</td>
</tr>
<tr>
<td>UC2</td>
<td>70%</td>
<td>0%</td>
<td>20%</td>
</tr>
<tr>
<td>UC3</td>
<td>31%</td>
<td>43%</td>
<td>14%</td>
</tr>
<tr>
<td>UC4</td>
<td>71%</td>
<td>25%</td>
<td>25%</td>
</tr>
<tr>
<td>UC5</td>
<td>60%</td>
<td>0%</td>
<td>20%</td>
</tr>
<tr>
<td>Average</td>
<td>61%</td>
<td>19%</td>
<td>21%</td>
</tr>
</tbody>
</table>

The large majority of irrelevant details in the intervention group are caused by description of the optional/none core FEs of the frames. For example: A step of normal event stating that “The Diner orders the food of his/her desire by careful selection” (by one of participant with frame). This step is completely irrelevant in given context which was triggered with FE Manner. Similarly, a step of normal event “The Kitchen staff prepares the meal in sequence requested by the customer”. This step was triggered with FE Time which is part of the Order Preparation use case and irrelevant in scenario of the Input order use case thus we have found that most of the irrelevant steps in use cases with frames are caused by none core FEs.

Another issue of causing irrelevant steps in use cases developed With frame is that some of the participants have identified only one core FE per step in a single use case (e.g., “The Diner calls the waiter to order the food ” whereas the FE Entity correspond to food, but the FE Diner relates to FE Customer but the participants linked one FE to one step only.

The average irrelevance for the three use cases of control group is 19%, 27%, and 59% respectively. Similarly, the average irrelevance for intervention group is 61%, 21%, and 54% respectively. This clearly represents that the use case Input order details,
developed with frame, has higher number of irrelevant steps (i.e. 61% vs 19%), while in other two use cases frames perform better, and less irrelevant steps are identified.

Thus, from this study, we can see that frame has made little improvement in this area, this could be caused by insufficient methodological support on use of frames for use case development provided to the intervention group members. Further research is necessary to resolve this issue.

4.3.3 Clarity

To assess clarity of use cases, one can look at their readability and structure. Here we assess if the conditions for the alternative flows are correctly specified and are at the right location, i.e., if the flows of various scenarios are clearly separated. Since we requested only details of the main flows to be provided for the use cases, we observe how the alternative flows are tangled with the main flow. The readability is not taken into account in this study as the linguistics-based readability metrics results are overly affected by the English language skills of the study participants.

The mixed steps of both groups are illustrated in table 4.3.3-1. In terms of mixed steps of intervention group; in Input Order use case only 1 out of 5 participants mixed the flows with average of 1% of total mixed steps, the Print Bill use case did not have any alternative flows mixed with the main flow, the Input Payment Details use case had also no alternative flow mixed with the main flow.

In terms of control group; in Input Order use case 3 out of 5 participants had mixed alternative flow steps into the main flow with average of 13% of mixed steps, the Print Bill use case did not have any alternative flows mixed with the main flow, in Input Payment Details use case again 2 out of 5 participants had mixed the flows with average of 11% of mixed steps.
<table>
<thead>
<tr>
<th>Use Case</th>
<th>Input order Details</th>
<th>Print Bill</th>
<th>Input Payment Details</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>With Frame</td>
<td>Without Frame</td>
<td>With Frame</td>
</tr>
<tr>
<td>UC1</td>
<td>0%</td>
<td>0%</td>
<td>0%</td>
</tr>
<tr>
<td>UC2</td>
<td>0%</td>
<td>33%</td>
<td>0%</td>
</tr>
<tr>
<td>UC3</td>
<td>6%</td>
<td>7%</td>
<td>0%</td>
</tr>
<tr>
<td>UC4</td>
<td>0%</td>
<td>25%</td>
<td>0%</td>
</tr>
<tr>
<td>UC5</td>
<td>0%</td>
<td>0%</td>
<td>0%</td>
</tr>
<tr>
<td>Average</td>
<td>1%</td>
<td>13%</td>
<td>0%</td>
</tr>
</tbody>
</table>

Thus, for our clarity criterion, the intervention group outperformed the control group for two use cases and performed at par for the third one.

### 4.4 Threats to Validity

It is also important to justify how valid the results are accrued from conducted experiment. The results should be adequate for the population that is considered to extract the sample and should also be generalized to a broader population. Three main types of threats to validity including an internal, an external, and a conclusion validity are discussed as follows:

#### 4.4.1 Internal

This experiment was carried out with students as subjects, and therefore there is no certainty of the extent to which the results could be generalized to industrial settings. Although the students have previously worked in requirements engineering assignments and developed use cases in various assigned tasks. In order to overcome the challenges of using students as subjects, a qualitative study has been carried out in
Chapter 5 by selecting experienced people with industrial and research background as subjects. Although, we aim at reducing the generalization threat further by verifying the results via experienced analysts’ interviews.

The current study uses a small sample size of dataset. Though this may be considered as an impediment to generalization it is technically desirable for different reasons. The small sample size can serve as a helpful pilot and exploratory study as it is often better to test a new research hypothesis in a small number of subjects first to take an initiative and later implement it on larger population. It can result in high efficiency in experimental design wherein results can be obtained easily and efficiently. It also reduces the experimental time and will need less resources. Further, with small sample size experiments can be carried out with less disruption as data can be processed quickly and consistently.

4.4.2 External

The subject system we selected for our experiment is a small system consisting of a small number of use cases. Therefore, it is not certain that the results are applicable to a considerably larger system with a greater number of use cases. Although, it was found beneficial to take an initiative with a small use case model consisting of a small number of use cases and later implement it on larger system.

In our experiment we requested only details of the main flows, actor, pre-condition and post condition. The details of alternative and extended use cases were not used in this study. This may also mean that the use case models in our experiment are not comparable to the use case models for real systems on all aspects. In order to obtain a detailed feedback about our hypothesis, very simple structures of use cases have been used to make it easy for students.

4.4.3 Construct Validity

To assess the clarity of use cases, one can look at their readability and structure. Since we requested only details of the main flows to be provided for the use cases and the alternate flow was not the part of this study, therefore the assessment of clarity may be invalid. To tackle this problem, we assessed if the conditions for the alternative flows are correctly specified and are at the right location,
i.e., if the flows of various scenarios are clearly separated. We observed how the alternative flows are tangled with the main flow.

4.5 Summary

This chapter proposes a vision that provides an initial evaluation of the proposition that using frames as a reference model when undertaking requirements specification, could improve the quality of use cases. The result of the study shows that qualities of use case developed with frame are complete and clear.

In terms of completeness the number of missing actors for use case (“Input Order”, “Print Bill” and “Input Order deal”) without frames are respectively 2, 6 and 4 times greater than the use cases with frame. Similarly, the number of failures of goal achievement in use cases without frame is also higher than the use cases with frame. The number of failures for selected three use cases without frame is 2, 3, 2 respectively and 0.2.0 for use case with frame.

In terms of clarity the average percentage of mixed steps for use case (“Input Order”, “Print Bill” and “Input Order deal”) without frame is 13%, 0%, 11% respectively while the average percentage of mixed steps for use case with frame is 1%, 0%, 0%.

From this study, we can conclude that FrameNet frame improve the quality of use cases in terms of completeness, consistency and clarity.
Chapter 5

5 FrameNet Frames Usage- A Think Aloud Activity

This chapter presents a qualitative study to determine how FrameNet frames can be used for requirement elicitation. Section 5.1 introduces research questions. The design and methodology of the study is outlined in Section 5.2. Section 5.3 presents the results and discussion. Section 5.4 evaluates the validity of the results achieved and is followed by summary of the chapter in Section 5.5.

5.1 Research Question and Hypothesis

The previous study demonstrated that FrameNet is capable of leading to improved requirements. In this chapter we present a study aimed to address the question on use of methodology. The study investigates why and in what respects FrameNet can be helpful for requirements elicitation. To avoid the problems posed by inexperienced undergraduates, this study uses a different cohort of more experienced participants.

To investigate how different participants, think and use frames, an empirical study was conducted with the following two research questions:

*RQ1*: How helpful is FrameNet for creating use cases?

*H1*: FrameNet frames provide support for identifying useful information.

*RQ2*: How is the FrameNet used by requirement analysts in creating the use cases?

*H2*: We anticipate that there are common usage patterns by participants.

5.2 Methodology

This section describes the approaches used for sampling, data collection, and training.

5.2.1 Sampling

Sampling was based on convenience sampling method [103, 104]. Personal contacts of author were utilized to contact people from a Computer Science background. The demographic and consent forms were sent to them to obtain their consent and to know about their experiences in use cases. Those with no basic experience of use cases were disregarded and only those subjects were selected who had worked with use cases. Along with convenience sampling, the snowball (or referral) sampling method [105,
was adopted where the existing study participants were asked to recruit further respondents from among their contacts. The participants were contacted via social media, skype, phone and their email addresses.

The number of targeted participants was 25. Eight of these had no experience in use cases, five of them did not reply, and two of them were not available in that particular time span that was fixed for our interview. This left us with ten subjects who were able to participate in the study.

The participants were divided into two groups. We tried to allocate a similar level of experience and academic qualifications to both groups (obtained from demographic information sheet see Appendix section C). In the field of software engineering, the application of such experiments where you have to group the people of same level education, experience is not an easy task because it involves different levels of knowledge, interpretation and degree of applicability [107].
5.2.2 Training and Tasks

This section provides the detail of training and martial provided to control and experimental group.

**With FrameNet**

i. The participants were provided with study information sheet, demographic information sheet and consent form.

ii. The participants were provided with PowerPoint slides, Frame details and other detailed outlines of the study a week before interviews.

iii. The participants were given a chance to ask questions about the study before interview took place.

iv. The experimental group was given fifteen minutes for a question/answer session.

v. Participants were asked to create use cases using frame and system description of selected case study.

vi. Participants were given 40 minutes to complete task.

vii. The participants were told that the use case could include element such as Actors, Precondition, Post-condition, Normal events, and Alternative events.

viii. Participants perform their task voluntarily.

**Without FrameNet**

i. The participants were provided with study information sheet, demographic information sheet and consent form.

ii. The participants were provided with PowerPoint slides and other detailed outlines of the study a week before interviews.

iii. The participants were given a chance to ask questions about the study before interview took place.

iv. The control group was given ten minutes for a question/answer session.

v. Participants were asked to create use cases using system description of selected case study.

vi. Participants were given 40 minutes to complete task.

vii. The participants were told that the use case could include element such as Actors, Precondition, Post-condition, Normal events, and Alternative events.

viii. Participants perform their task voluntarily.
5.2.3 Data collection

In this study, we paid direct attention to the participant’s thinking and understanding about the usage of frames for requirements elicitation. Since this study is based on exploring individuals’ thought processes, the Think Aloud protocol is adapted to capture the ideas of participants. The Think Aloud protocol has previously been used for requirement elicitation in [108-110] and was considered a cheap, robust, flexible, convenient and easy method to learn about the participants’ thoughts.

The aim of using Think Aloud was to record the participants’ thoughts whilst they explain how they identify the actors, and from where they get their ideas (i.e. whether the idea comes from own experience, frames or textual description of overview). The participants were asked to express their thoughts loudly by explaining how they performed different activities. The interviews were taken as audio recordings.

5.2.4 Data Analysis

In this section, we describe the process of data coding and data analysis for conducted interviews. Coding is used for the conceptual abstraction of data and its reintegration as a theory [111]. There are three coding approaches used for qualitative data analysis such as open, axial and selective [112]. In this study, the open and selective coding schemes are used to analyze the interviews. In open coding process the data (written text) is read several times to discover ideas, concepts and theories. In this coding approach the data is coded in every way possible till the analyst gets valuable information. The significance of the open coding is that concepts are emerged from the raw data and are later grouped into conceptual categories. Such coding can be done either manually or using any software tool. In selective coding process the sub categories of code are categorized under main category [112].

In our case, we split up and analyzed the data first through open coding to choose a core category (running a data open), and then a selective coding approach is applied by relating all other categories to the chosen core category. For example, the FE ‘Money’ is chosen as main code (using open code scheme), and all those steps triggered by FE Money in same sense are grouped under the same code ‘Money’ (selective code). For instance, different participants used statements such as “waiter takes the money”, “Diner paid amount for food”, and “Waiter collects the payment from the diner”. Because these all involved LUs ‘paid’, ‘payment’ and FE ‘money’, all are selected and
grouped under the category of code “Money”. The codes are representation of elements (information) triggered by participants with and without frame.

The use cases created by participants are evaluated to check that the elements (information) extracted by both groups’ participants are valid in the given scenario or not. The transcription was organized into a sequence of steps. Contents of use cases and verbal interviews were reviewed to count the number of valid and invalid elements. The valid/invalid decisions were applied to all aspects of the use-cases produced by the participants, including actors, pre-conditions, post-conditions, normal events, and alternative events.

In terms of RQ1, the number of valid and invalid elements elicited with and without frames was compared. A set of statistical tests are applied on sample data to evaluate the hypothesis. For each sample, the Minimum value, Median, Lower Quartile, Upper Quartile and Maximum values are calculated using Graphpad Prism tool [113]. The Lower Quartile and Upper Quartile are represented by 25% percentile and 75% Percentile respectively.

In case of RQ2, the sequences of steps followed by the participants to create use cases with frame are analyzed. It is also evaluated, how did the different aspects (Actor, Precondition, Post-condition, Normal Events and Alternative Events) of use cases were generated? Was it either by system overview, by selected frames or by the participants’ own experience/own thinking?

5.3 Results

This section illustrates the results obtained for stated research questions. The term information, elements and codes are representing same idea and used alternatively.

5.3.1 Research question 1

RQ1: How helpful is FrameNet for creating use cases?

The data obtained for RQ1 is illustrated in table 5.3.1.1-1, table 5.3.1.1-2, table 5.3.1.1-3, table 5.3.1.1-4. The table values “1” and “0” represent the triggered and untriggered status of elements respectively. The prefix “PF” represents participants who used frames, whereas “P” represents the participants without frames. There were 25 and 34 elements observed for use case “Input Order” and “Input Payment Detail” respectively (details are available in Appendix C).
5.3.1.1 Use Case 1

Table 5.3.1.1-1 represents the performance of each participant eliciting valid elements with and without frames. The total number of valid elements for use case 1 is 17. Here it can be seen that the highest number of valid entries is triggered by participants with frames, where PF3, PF2 and PF5 triggered 15, 14 and 12 numbers of valid entries respectively (grey shaded columns).

**Table 5.3.1.1-1 Valid elements elicited by each participant for use case 1**

<table>
<thead>
<tr>
<th>Valid set of elements triggered by participants for use case1 “Input order”</th>
<th>Participants with frame</th>
<th>Participants without frame</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>PF1</td>
<td>PF2</td>
</tr>
<tr>
<td>Supplier as Staff(kitchen/Bar)</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>Supplier as waiter</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>Customer as diner</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>Beneficiary as diner (diner group)</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>Means</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Medium (pc/iPad)</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>Time</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>Entity as item/food</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>Waiter as actor</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>Kitchen staff (Supplier) as actor</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>Item is available</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>Quantity</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>Order is given</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Order time is not recorded</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>Item is not available</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>Change of order or alternative order</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>Order is accepted/done or received by staff</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>Total</td>
<td>11</td>
<td>14</td>
</tr>
</tbody>
</table>
Table 5.3.1.1-2 represents the frequency of valid entries triggered with and without frames. The total number of valid entries with and without frames for all elements is 66 and 54 respectively. The results indicate that the participants who used frames have triggered a higher number of valid entries with frames as compared to those without frames.

Table 5.3.1.1-2 Number of times the participants triggered each valid element for use case 1

<table>
<thead>
<tr>
<th>Valid set of elements for use case 1 “Input order”</th>
<th>Number of participants who selected each element with frames</th>
<th>Number of participants who selected each element without frames</th>
</tr>
</thead>
<tbody>
<tr>
<td>Supplier as Staff(kitchen\Bar)</td>
<td>5</td>
<td>5</td>
</tr>
<tr>
<td>Supplier as waiter</td>
<td>4</td>
<td>3</td>
</tr>
<tr>
<td>Customer as diner</td>
<td>4</td>
<td>4</td>
</tr>
<tr>
<td>Beneficiary as diner (diner group)</td>
<td>4</td>
<td>5</td>
</tr>
<tr>
<td>Means</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>Medium (PC/iPad)</td>
<td>5</td>
<td>5</td>
</tr>
<tr>
<td>Time</td>
<td>5</td>
<td>2</td>
</tr>
<tr>
<td>Entity as item/food</td>
<td>5</td>
<td>5</td>
</tr>
<tr>
<td>Waiter as actor</td>
<td>5</td>
<td>4</td>
</tr>
<tr>
<td>Kitchen staff as actor</td>
<td>5</td>
<td>3</td>
</tr>
<tr>
<td>Item is available</td>
<td>5</td>
<td>5</td>
</tr>
<tr>
<td>Quantity</td>
<td>1</td>
<td>4</td>
</tr>
<tr>
<td>Order is given</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>Order time is not recorded</td>
<td>3</td>
<td>0</td>
</tr>
<tr>
<td>Item is not available</td>
<td>5</td>
<td>3</td>
</tr>
<tr>
<td>Change of order or alternative order</td>
<td>3</td>
<td>1</td>
</tr>
<tr>
<td>Order is accepted/done or received by staff</td>
<td>5</td>
<td>4</td>
</tr>
<tr>
<td>Total</td>
<td>66</td>
<td>54</td>
</tr>
</tbody>
</table>

Table 5.3.1.1-3 represents the performance of each participant eliciting invalid elements with frame and without frames. The total number of invalid elements for use case 1 is 8. Here it can be seen that the highest value (3) of invalid entries is triggered
by participants from both groups (PF1 and P2) while P1 triggered none of the invalid information.

Table 5.3.1.1-3 Invalid elements elicited by each participant for use case 1

<table>
<thead>
<tr>
<th>Invalid set of elements triggered by participants for use case 1 “Input order”</th>
<th>Participants with frame</th>
<th>Participants without frame</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>PF1</td>
<td>PF2</td>
</tr>
<tr>
<td>Customer as waiter</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>Beneficiary as Supplier</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Manner (manner as special order)</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Manner (as carefully)</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Type of food</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Dinner as actor</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>Cashier as an actor</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Technical Problem</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>Total</td>
<td>3</td>
<td>1</td>
</tr>
</tbody>
</table>

Table 5.3.1.1-4 represents the frequency of invalid entries triggered with and without frames. The total number of invalid entries with and without frames for all elements is 9 and 6 respectively. It can be seen that the use of FrameNet frames for requirement elicitation resulted in higher number of invalid entries as well.
Table 5.3.1.1-4 Number of times the participants triggered each invalid element for Use Case 1

<table>
<thead>
<tr>
<th>Invalid set of elements for use case 1 “Input order”</th>
<th>Number of participants who selected each element with frames</th>
<th>Number of participants who selected each element without frame</th>
</tr>
</thead>
<tbody>
<tr>
<td>Customer as waiter</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>Beneficiary as Supplier</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>Manner (manner as special order)</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>Manner (as carefully)</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Type of food</td>
<td>0</td>
<td>2</td>
</tr>
<tr>
<td>Dinner as actor</td>
<td>3</td>
<td>2</td>
</tr>
<tr>
<td>Cashier as an actor</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>Technical Problem</td>
<td>3</td>
<td>1</td>
</tr>
<tr>
<td>Total</td>
<td>9</td>
<td>6</td>
</tr>
</tbody>
</table>

The result of statistical analysis for valid and invalid elements triggered with and without frame is shown in table 5.3.1.1-5

Table 5.3.1.1-5 Statistical analysis of use case 1

<table>
<thead>
<tr>
<th>Use Case 1</th>
<th>Valid</th>
<th></th>
<th>Invalid</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>With Frame</td>
<td>Without Frame</td>
<td>With Frame</td>
<td>Without Frame</td>
</tr>
<tr>
<td>Number of Entries</td>
<td>17</td>
<td>17</td>
<td>8</td>
<td>8</td>
</tr>
<tr>
<td>Minimum</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>25% Percentile</td>
<td>3</td>
<td>1.5</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Median</td>
<td>5</td>
<td>4</td>
<td>1</td>
<td>0.5</td>
</tr>
<tr>
<td>75% Percentile</td>
<td>5</td>
<td>5</td>
<td>2.5</td>
<td>1.75</td>
</tr>
<tr>
<td>Maximum</td>
<td>5</td>
<td>5</td>
<td>3</td>
<td>2</td>
</tr>
<tr>
<td>Total</td>
<td>66</td>
<td>54</td>
<td>9</td>
<td>6</td>
</tr>
</tbody>
</table>

101
Figure 5.3-1 Statistical analysis of use case 1

Figure 5.3.1.1-1 represents the statistical analysis of data obtained for use case 1. The x-axis represents the valid and invalid information triggered by participants of both groups. The grey and white plots represent the information triggered with and without frames respectively. In terms of valid information, the values of maximum and upper quartile for participants with and without frames are the same, but the values of minimum, lower quartile and median with frames are higher than the value without frame. Thus, it implies that frames are helpful for obtaining valid information.

In terms of invalid information, figure 5.3.1.1-1 shows that the values of minimum and lower quartile are 0 with and without frame, while the values of maximum, median and upper quartile with frame are higher than the value without frame.

Unpaired t-test is used to identify significant difference of valid and invalid information. The P-value for valid information is 0.224 indicating that there is no significant difference between information elicited with and without frames. Similarly, there is no significant difference between invalid information elicited with frame and without frame as P=0.4994.
5.3.1.2 Use Case 2

Table 5.3.1.2-1 represents the performance of each participant eliciting valid elements with and without frames for use case 2. The total number of valid elements for use case 2 is 24. Here it can be seen that the highest number of valid entries is triggered by the participant with frame (PF5) and the lowest number of valid information is triggered without frame by P1. The lowest number of valid entries by frame group (PF1) is higher (13) than the highest number (12) of the group without frames (P2).

Table 5.3.1.2-1 Valid element elicited by each participant for use case 2

<table>
<thead>
<tr>
<th>Valid set of elements triggered by participants for use case2 “Input Payment details”</th>
<th>PF1</th>
<th>PF2</th>
<th>PF3</th>
<th>PF4</th>
<th>PF5</th>
<th>P1</th>
<th>P2</th>
<th>P3</th>
<th>P4</th>
<th>P5</th>
</tr>
</thead>
<tbody>
<tr>
<td>Buyer as diner</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>Seller as waiter</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>Seller as Management staff</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>Unit (Currency)</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Money</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>Rate as price per item</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>Good (Dishes/item)</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>Means (Card/cash)</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>Time</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Waiter as actor</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>Management as actor</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Tip</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Discount</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Diner receives the bill</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>Print receipt</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>Successful Completion of payment</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>Successful Completion of Order</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>Wrong information</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>Unit: wrong currency</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Technical Problem</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>Means: unacceptable mode of payment</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Payment time is not recorded</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Don’t have Money</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Change of money is given</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>Total</td>
<td>13</td>
<td>18</td>
<td>15</td>
<td>16</td>
<td>19</td>
<td>8</td>
<td>12</td>
<td>10</td>
<td>10</td>
<td>9</td>
</tr>
</tbody>
</table>
Table 5.3.1.2-2 represents the frequency of valid entries triggered with and without frames for use case 2. The total number of valid entries with and without frames for all elements is 88 and 54 respectively. Thus, use case 2 also suggests that frames are helpful for requirements elicitation and in obtaining relevant information.

**Table 5.3.1.2-2 Number of times the participants triggered each valid element for use case 2**

<table>
<thead>
<tr>
<th>Valid set of elements for use case 2 “Input payment detail”</th>
<th>Number of participants who selected each element with frames</th>
<th>Number of participants who selected each element without frames</th>
</tr>
</thead>
<tbody>
<tr>
<td>Buyer as diner</td>
<td>5</td>
<td>4</td>
</tr>
<tr>
<td>Seller as waiter</td>
<td>5</td>
<td>5</td>
</tr>
<tr>
<td>Seller as Management staff</td>
<td>4</td>
<td>4</td>
</tr>
<tr>
<td>Unit (Currency)</td>
<td>4</td>
<td>0</td>
</tr>
<tr>
<td>Money</td>
<td>5</td>
<td>4</td>
</tr>
<tr>
<td>Rate as price per item</td>
<td>4</td>
<td>3</td>
</tr>
<tr>
<td>Good (Dishes/item)</td>
<td>5</td>
<td>4</td>
</tr>
<tr>
<td>Means (Card/cash)</td>
<td>5</td>
<td>1</td>
</tr>
<tr>
<td>Time</td>
<td>5</td>
<td>0</td>
</tr>
<tr>
<td>waiter as actor</td>
<td>5</td>
<td>4</td>
</tr>
<tr>
<td>Management as actor</td>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td>Tip</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>Discount</td>
<td>2</td>
<td>1</td>
</tr>
<tr>
<td>Diner receives the bill</td>
<td>4</td>
<td>4</td>
</tr>
<tr>
<td>Print receipt</td>
<td>2</td>
<td>1</td>
</tr>
<tr>
<td>Successful Completion of payment</td>
<td>5</td>
<td>2</td>
</tr>
<tr>
<td>Successful Completion of Order</td>
<td>4</td>
<td>4</td>
</tr>
<tr>
<td>wrong information</td>
<td>0</td>
<td>2</td>
</tr>
<tr>
<td>Unit: wrong currency</td>
<td>4</td>
<td>0</td>
</tr>
<tr>
<td>Technical Problem</td>
<td>2</td>
<td>1</td>
</tr>
<tr>
<td>Means: unacceptable mode of payment</td>
<td>5</td>
<td>0</td>
</tr>
<tr>
<td>Payment time is not recorded</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>Don’t have Money</td>
<td>3</td>
<td>0</td>
</tr>
<tr>
<td>Change of money is given</td>
<td>0</td>
<td>2</td>
</tr>
<tr>
<td>Total</td>
<td>81</td>
<td>49</td>
</tr>
</tbody>
</table>
Table 5.3.1.2-3 represents the performance of each participant eliciting invalid element with and without frames. The total number of invalid entries coded for use case 2 is 10. Here we can see that the highest value (4) of invalid entries is triggered by participant (P5) without frames.

Table 5.3.1.2-3 Invalid element elicited by each participant for use case 2

<table>
<thead>
<tr>
<th>Invalid set of elements triggered by participants for use case 2 “Input Payment details”</th>
<th>PF1</th>
<th>PF2</th>
<th>PF3</th>
<th>PF4</th>
<th>PF5</th>
<th>P1</th>
<th>P2</th>
<th>P3</th>
<th>P4</th>
<th>P5</th>
</tr>
</thead>
<tbody>
<tr>
<td>Place</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Purpose</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Rate as unit/quantity</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Explanation</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Frequency</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>System as actor</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>Cashier as actor</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>Diner as actor</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Payment is calculated</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>Quantity</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>Total</td>
<td>1</td>
<td>2</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>2</td>
<td>0</td>
<td>1</td>
<td>4</td>
</tr>
</tbody>
</table>

Table 5.3.1.1-4 represents the frequency of invalid entries triggered with and without frames for use case 2. The total number of invalid entries with and without frames for all elements is 5 and 7 respectively.
Table 5.3.1.2-4 Number of times the participants triggered each invalid element for Use Case 2

<table>
<thead>
<tr>
<th>Invalid set of elements for use case 2 “Input payment detail”</th>
<th>Number of participants who selected each element with frames</th>
<th>Number of participants who selected each element without frames</th>
</tr>
</thead>
<tbody>
<tr>
<td>Place</td>
<td>4</td>
<td>1</td>
</tr>
<tr>
<td>Purpose</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Explanation</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>Frequency</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Rate as unit/quantity</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Diner as actor</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>System as actor</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>Cashier as actor</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>Quantity</td>
<td>0</td>
<td>2</td>
</tr>
<tr>
<td>payment is calculated</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>Total</td>
<td>5</td>
<td>7</td>
</tr>
</tbody>
</table>

Table 5.3.1.2-5 Statistical analysis of use case2

<table>
<thead>
<tr>
<th>Use Case 2</th>
<th>Valid Information</th>
<th>Invalid Information</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>With Frame</td>
<td>Without Frame</td>
</tr>
<tr>
<td>Number of Information</td>
<td>24</td>
<td>24</td>
</tr>
<tr>
<td>Minimum</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>25% Percentile</td>
<td>2</td>
<td>0.25</td>
</tr>
<tr>
<td>Median</td>
<td>4</td>
<td>2</td>
</tr>
<tr>
<td>75% Percentile</td>
<td>5</td>
<td>4</td>
</tr>
<tr>
<td>Maximum</td>
<td>5</td>
<td>5</td>
</tr>
<tr>
<td>Total</td>
<td>81</td>
<td>49</td>
</tr>
</tbody>
</table>
Figure 5.3-2 Statistical analysis of use case 2

Figure 5.3.1.2-1 represents the statistical analysis of data obtained for use case 2. The x-axis represents the valid and invalid information (element) triggered by participants of both groups. The grey and white plots represent the information triggered with and without frame respectively.

The P-value for valid elements is 0.0112, which indicates a significant difference between information elicited with frame and without frame. On the other hand, there does not appear to be a significant difference between invalid entries elicited with frame and without frame as the P-value is 0.6652. Thus, the results of use case 2 indicate that frames perform a positive role in requirements elicitation. Hence a set of more complete requirements can be obtained by using FrameNet frames as it provides a relevant information about particular concept.
5.3.2 Research Question 2

**RQ2: How is the FrameNet used by requirement analysts in creating the use cases?**

This section describes the steps created by different participants during use case elicitation with frames. The short terms used in this section are as follows: Pre-Condition of use case (PreCon), Post-Condition of use case (PostCon), Normal event of use case (NE), Alternative events of use case (Alt), Frame Elements of frame (FEs), Frame definition and example (Frame), Textual description of system overview (Overview), own idea from experience (OwnThink).

5.3.2.1 Use Case 1

Table 5.3.2.1-1 illustrates the different steps that took place during use case elicitation with frames by each participant. Although there are no identical patterns followed by all participants, there are several similarities. For example: the steps Frame → FEs, FEs → LnkFEs, LnkFEs → FEs, FEs → NEs, NEs → FEs, and Overview → Alt are followed by all five participants. The transactions from FEs → Actors, Actors → FEs, Overview → NE, FEs → PreCon, PrCon → FEs are followed by four participants. There are only four transactions which are unique to individual participants, and these are PreCon → Overview, Overview → PreCon, Overview → PostCon, Alt → FEs. We can see that pattern 2 and pattern 4 have at least one common step with other patterns. The colored cells in table 5.3.2.1-1 represent the common steps in each pattern.
Table 5.3.2.1-1 Patterns of participants with frame

<table>
<thead>
<tr>
<th>Transactions</th>
<th>Pattern 1</th>
<th>Pattern 2</th>
<th>Pattern 3</th>
<th>Pattern 4</th>
<th>Pattern 5</th>
</tr>
</thead>
<tbody>
<tr>
<td>FEs → Actors</td>
<td>3</td>
<td>1</td>
<td>2</td>
<td>0</td>
<td>3</td>
</tr>
<tr>
<td>Actors → FEs</td>
<td>3</td>
<td>1</td>
<td>2</td>
<td>0</td>
<td>3</td>
</tr>
<tr>
<td>Overview → Actors</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>3</td>
<td>0</td>
</tr>
<tr>
<td>Frame → FE</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>FEs → LnkFEs</td>
<td>3</td>
<td>3</td>
<td>4</td>
<td>4</td>
<td>4</td>
</tr>
<tr>
<td>LnkFEs → FEs</td>
<td>3</td>
<td>3</td>
<td>4</td>
<td>4</td>
<td>4</td>
</tr>
<tr>
<td>FEs → NE</td>
<td>3</td>
<td>3</td>
<td>5</td>
<td>1</td>
<td>4</td>
</tr>
<tr>
<td>NE → FEs</td>
<td>1</td>
<td>4</td>
<td>3</td>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td>Overview → NE</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td>NE → Overview</td>
<td>2</td>
<td>1</td>
<td>2</td>
<td>1</td>
<td>2</td>
</tr>
<tr>
<td>OwnThink → NE</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>NE → OwnThink</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>FEs → PreCon</td>
<td>1</td>
<td>2</td>
<td>2</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>PrCon → FEs</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>PreCon → Overview</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>OwnThink → PreCon</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>FE → PostCon</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>PostCon → FEs</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>OwnThink → PostCon</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>Overview → PostCon</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Overview → Alt</td>
<td>2</td>
<td>1</td>
<td>3</td>
<td>2</td>
<td>1</td>
</tr>
<tr>
<td>Alt → Overview</td>
<td>1</td>
<td>1</td>
<td>2</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Source</td>
<td>Alt 1</td>
<td>Alt 0</td>
<td>FE 0</td>
<td>FE 1</td>
<td>Overview 0</td>
</tr>
<tr>
<td>----------------</td>
<td>-------</td>
<td>-------</td>
<td>------</td>
<td>------</td>
<td>------------</td>
</tr>
<tr>
<td>OwnThink</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>FEs</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>Alt</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Overview</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
</tbody>
</table>
Figure 5.3-3 schematic representation of PF 1 with frame

Figure 5.3.2.1-1 is a schematic representation of sequence adopted by PF1 for use case 1 with frames. The schematic representation of sequence adopted by remaining participants (PF2-PF5) is available in Appendix C. In this sequence, it is observed to know how the frames are used to elicit different aspects (Actor, Pre-condition, Post-condition, Normal Events and Alternative Events) of use cases.

PF1 started with Overview and then read Goal and Description of use case. After reading the frame definition the FEs were used to identify actors. The FEs were further utilized to retrieve NE1, NE2, NE4, PreCon and PostCon. The Overview was checked again to elicit NE3, Alt1 and Alt2. The participants also used their own thinking and understanding (apart from frames and system overview) to write down different steps
of use cases. The steps triggered by the participants using own thinking or previous experience are grouped under Own Thinking category. For example, in figure 5.3.2.1-1 the Alt3 was elicited with participant’s own understanding.

5.4 Discussion

In terms of RQ1, frames performed a better role in requirements elicitation for both use cases. The total number of elements of use case 1 is 17. We can see that the information triggered with frames such as “Supplier as kitchen/bar staff”, “Customer as dinner” and “Medium as PC/iPad” is also elicited by some of the participants without frames, which indicates that frames are generally aligned with the way that stakeholders perceive the world, and are also understandable to the requirements practitioners (detailed in table 5.3.1.1-1).

FEs appear to be helpful, to identify more relevant actors for both use cases as compared to use cases without frame. For example, the FE Supplier was identified as a valid actor by all participants using frames and was identified by only three of the participants who did not use frames (detail in table 5.3.1.1-1).

The participants also linked the information described in the textual description of system overview with relevant FEs. For example, the FEs “Medium” “Entity” and “Supplier” are correctly linked to PC, Food and Staff member respectively by all participants (PF1 to PF5). Similarly, the FEs “Customer” and “Supplier” are correctly linked to the Diner and Waiter respectively by four participants (PF2 to PF5). Some of the missing information in system overview is also elicited with frames. For example, “Time” is one of the important factors in making an order but was not identified by any participants who created use cases without frames.

We can see that frames appear to help in eliciting valid information or requirements, but also triggered some invalid information for use case 1 as described in table 5.3.1.1-3. These results might be affected by the poor performance of individual participants. For example, PF1 uses frames but has elicited the lowest rate (11) of valid and highest rate (3) of invalid information.

On other hand, the result of use case 2 is different than the use case 1 as more valid and less invalid information was elicited by participants using frames. The total number of elements for use case 2 is 24. It shows that frames perform better in elicitation of use
case 2. The frames performed a better role in elicitation valid set of requirements in both use cases.

In terms of RQ2, there are no interactions that are identical from start to finish. However, we did identify several commonalities followed by all participants. For example, identification of actors from frames, reading frame definitions and examples, linking FEs to different parts of a use case and textual description of system.

The actors are identified with Overview by only one participant. One of the participants identified one actor with frames and one from the overview. Three of the participants have identified actors with frames. Thus, most of the actors appear to be identified with frames. The actors identified from the overview by some participants are identified with FEs by others. For example, in pattern 2 and pattern 4 the actor “Waiter” is identified from Overview and same actor is identified with FE “Supplier” in pattern 3 and pattern 5. There are many FEs which are commonly used in all patterns, for example the FEs “Supplier”, “Customer”, “Entity”, “Medium”, and “Time”. Most of the FEs were used by all participants for same type information. For example, in the “Request-Entity” frame, the FE “Supplier” was considered as a staff member, and “Entity” as an item or food to be delivered. The FE “Customer” was used as diner by all participants but PF1 use it as Waiter which is a wrong concept.

5.5 Threats to validity

This section describes the threats and validation of threats. Internal and external threats are used to evaluate the validity of a research study and procedure [114]. The external validity is the extent to which the results of a study can be generalized to other situations and other people. Internal validity shows how well an experiment is done. It evaluates that whether the effects observed in a study (method) are due to the manipulation of the independent variable and or due to some other factor.

5.5.1 Internal threats

Prior experience in use-case modelling could bias the results; more experienced participants could produce better specifications irrespective of their use of frames. To attenuate this problem, we tried to select people with same experience and qualification and equally distributed them in two groups. At same level, the experience in academics, teaching and industry was considered that can also affect the results.
The participants were provided a chance to ask questions and to answer about any confusing point of this study just fifteen minutes before interview took place. To learn about frames and write a use case description in the wider industrial context, the given fifteen minutes might not prove enough in practical sense, but the timetabling pressure did not allow more time to provide more training. To minimize this threat, the participants were handed over a set of preselected frames. The participants did not have to go through the process of frame selection and other details such as LUs, frame to frame relationship, required for frame selection. This factor is controlled as they were already provided with selected frames for the corresponding use cases and were not asked to search for frames and understand.

In current study we are unable to quantify weight of each information (added from frame) to determine how important this information is? In this regard, the requirements triggered by participants were evaluated by the author of the study to check whether these requirements are valid or not in given scenario, which introduces a risk of bias. To reduce this risk, quantitative study is undertaken in upcoming chapter (chapter 6), where participants are asked to rate each information added from frame.

5.5.2 External Validity

In this study the participants were asked to eliciting requirements for only two use cases due to time constraints. Therefore, we do not have prior knowledge about applicability or non-applicability of the results to a large system involving a greater number of use cases, but it may make easy for participants to write the use cases correctly.

5.6 Summary

This study is carried in this chapter to determine how helpful the frames are in requirements elicitation and how the different participants have used frame for use case development. In this regard, a verbal interview of experienced analysts was recorded using the Think Aloud approach. The findings of this study indicate that frames can help to elicit more valid elements, a greater number of valid actors and some missing information. In use case 1 there is no significant difference between use cases developed with and without frame but the results for use case 2 indicate a significant difference between use cases developed with and without frames. This is because such
frames appear to be generally aligned with the way that stakeholders perceive the world and are also equally understandable to the requirement practitioners.

For RQ2 we have identified some of the common steps in all patterns and also determined that the steps triggered with frames are mostly common among all participants such as Frame→FEs, FEs→LnkFEs, LnkFE→FEs, FEs→NEs and NEs→FEs.
Chapter 6

6 Survey

Empirical software engineering has gathered much significance over a couple of decades for studying the maintenance and development of large, software-intensive systems [115, 116]. The selection of an appropriate strategy for study depends mainly on the purpose and conditions of the investigation [117]. Qualitative data consists of words, while quantitative data consists of numbers. Quantitative studies produce numbers that should be interpreted before conclusions may be drawn. Survey is one of the type of quantitative research [118].

In current study, the user survey approach was adopted to describe and quantify the characteristics of a large population [119]. Survey is a best approach than any other research method as it evaluates thoughts, opinions, and feelings of many people by studying a subset of them. Original data and information of the survey method leads the researcher’ work towards unbiasedness, efficiency and more productivity. In a survey, the responses of the respondents are quantitatively analyzed by numeric variables (e.g. how many; how much; or how often) to draw conclusion. It aims to determine how the participants define meaning from their surroundings and how their behavior is influenced by this meaning? It, therefore, quantifies the opinions, attitude, behavior and other defined variables.

In this chapter, a broader range of study is conducted among expert users of software and agriculture domain Section 6.1 presents the aims and objectives. Section 6.2 discusses the research question of the study. Section 6.3 illustrates the methodology of study. Section 6.4 presents results and discussion. The threats to validity are presented in Section 6.5. Summary of the chapter has been given in section 6.6.

6.1 Aims and objectives

The findings of Chapter 5 indicate the positive role of FrameNet in requirements elicitation. It was important to conduct a wider empirical study to obtain feedback about the information added in use cases with frame support. Previous studies have mostly been conducted among professional participants working in requirements and software
development field. This posed a challenge to the validity of the results, making it necessary to consider the views of the non-technical participants because of their involvement, alongside developers, in requirement elicitation. As mentioned earlier that past studies in this area remained qualitative in nature reflecting that although FrameNet frames to be helpful (as discussed in Chapter 5), however, their helpfulness could not be quantified. The aim of this study is to understand if the experts using the tool (FrameNet frames) considered frames to be supportive for the purpose of requirements elicitation. It involved a large number of participants with high experience to determine which step of use cases are especially useful in a given scenario of GMS (summary in Appendix A). The participants of this study were, therefore, asked to indicate how useful the steps in the use cases were; including steps that have been elicited to complement the previously developed use case and ignored the steps that pre-existed in the previous version of the use case.

6.2 Research Questions

This study has obtained the views of gardening experts (agricultural domain) and software development experts in order to understand the relevance and usefulness of the given information.

For this purpose, a survey was conducted to explore the scoring of additional steps (added from FrameNet frames in use cases) and determine which steps are considered relevant and useful in the given scenario of GMS.

The survey investigated the usefulness of FrameNet frames in the requirements elicitation, specifically in use case development.

The overall research questions for this particular study were:

**RQ 1:** How relevant/useful are the additional steps added from frames for a given system?

The survey was intended to investigate whether users found the additional added steps useful and relevant?

As discussed earlier, the views of non-technical participants are considered important for the validity of this study, therefore, the response/views of participants experienced in agricultural domain are considered as important as those of software developers.

**RQ 2:** Is the additional information provided by FrameNet equally useful by participants having experience in gardening and software development?
6.3 Methodology

This section describes the procedure of survey design and implementation, sampling of population, data collection and data analysis.

To make the procedure and results more reliable, a pilot test was carried out. The participants of pilot test were selected from software and agriculture domain. Before the distribution of questionnaire, a draft was initially sent to two participants from each domain (software development and agriculture). While conducting the test, an effort was made to keep the questions/words of the questionnaire simple and suitable to avoid confusion and bias. Consent was received from participants prior to the completion of the survey’s questionnaire at the pilot stage.

The questionnaire was easy to understand for the participants from the software development domain who provided useful comments on the sequence of statements used in the questionnaire. The participants from the agricultural domain, on the other hand, faced difficulty in understanding the terminologies used in the questionnaire such as “how useful some of the steps in the use case, elicitation of requirements for use cases, and interacting with the program”.

The arrangement of the questions was not altered except few small changes in the organization/structure of the sentences and the conversion of technical terminologies into general ones. The pilot study was also useful in confirming the time it would take to complete the survey. The specified time for the survey was 15-20 minutes, however, pilot participants were able to complete the survey in 12 -15 minutes. The resulting questionnaire was reviewed and modified and further verified by two more users on from each domain.

6.3.1 Sampling

The target population for this survey was very precise. The survey mainly applied to the interested participants that were experts in GMS, therefore, it was important to give consideration to how the participants for this particular study were selected.

- **Agriculture domain:** The population from Agriculture domain, consisted of professional gardeners, some part-time gardeners, landlords, research students and
academic staff from Agriculture and Biotechnology departments of the universities in Pakistan who were engaged in growing different plants for their research.

- **Software development domain:** The population from software development domain consisted of research student, academic staff and professional software developers working in different institutes and software companies.

By using convenience sampling method, I utilized my personal contacts/links to contact the initial potential survey respondents who consented to participate in the survey. Along with convenience sampling [103], the snowball or referral sampling method [105] was adopted where the participants of the existing study were requested to recruit future respondents from among their contacts. In this method, the participants “with whom the contact has already been made used their social networks to refer the researcher to other people who could potentially participate in or contribute to the study” [120]. Snowball sampling is, therefore, a useful technique in order to increase the number of participants especially when the target population is that precise as is the case in this study. The initial participants of this particular study were, therefore, asked to forward the survey to other individuals or help the researcher in identifying individuals in their network who do possess interest in GMS.

Participants from both groups were contacted through telephone, social media and e-mails. As the number of the participants has a direct impact on the reliability and consistency of the survey, a sample size of 48 participants was chosen for this purpose.

### 6.3.2 Questionnaire & Protocol

Questionnaires have been used as a data collection tool to obtain an overall measure of the opinions and attitudes of the participants. It is a quick and simple method of gathering sound feedback from participants.

The questionnaire consists of the following four parts.

#### 6.3.2.1 Introduction

An introductory overview of the survey is given in this section. It explains the objective, tasks, contents and the approximate time required for the completion of survey.

#### 6.3.2.2 Consent

In this section, the consent of the users is obtained, and they are given the freedom to withdraw their consent at any time before the final submission.
6.3.2.3 Background

This section investigates the area and level of user experience. Users are asked to rate their experience in software development within the given Likert scale from 1 (No experience) to 5 (Very Experienced). This section consisted of two questions. The first question is asked from participants belonging to software development area and the second question is asked from participants belonging to agriculture background. The same rate of scale is distributed in both questions (i.e. 1-5).

6.3.2.4 Selected use cases

In this section, the structured nature of the questionnaire is organised to collect responses of users about the impact of the FrameNet frame on use case. In this regard, eight use cases are selected from GMS [102]. These use cases were previously developed by a requirement analyst. Each of the use cases has added few additional steps from frame. The five-points Likert scale is used to capture feedback of the respondents. Respondents were asked to rate the listed steps within the use case on a scale of 1 (very irrelevant) to 5 (very relevant) according to the usefulness of steps in the given scenario. The steps which pre-existed in the previous version of the use case are not evaluated. The following rates are used:

Very Irrelevant (VI) = 1, Irrelevant (I)=2, Neutral (N)=3, Relevant (R) =4, Very Relevant (VR)=5,

6.3.3 Data collection

After the survey was posted online for the participants, two weeks’ time was fixed for survey returns. During the collection of data, its authenticity and genuineness was checked through the preliminary analysis of the collected responses whose results from soft development domain proved very reasonable and sound but the response of the participants from agricultural domain were slow and less in number. Therefore, personal contacts were used to approach participants dealing with plants in Bio technology departments at various universities in Pakistan and the time span for survey was extended to three weeks. A total of 49 participants participated in the online survey and completed it successfully. The number of the respondents who left the survey incomplete was not calculated as this record was not compiled by the survey tool used. Google form website was used for the collection and storage of the data.
6.3.4 Data analysis

After the collection of data in the earlier step, the data-based analysis in order to draw conclusion was required. Thus, the collected data was analysed descriptively in order to answer the research questions of this study.

The obtained data is analyzed in two different ways for the stated research questions.

In terms of RQ1, the frequency of each rate is evaluated for every step (added from frames) using COUNTIF formula in Excel sheet. There are a total of forty eight steps (Q1-Q48, see detail in Appendix D) where each user rates every step from 1-5 (VI,...VR). For each set of information, the Minimum value, Median, Lower Quartile, Upper Quartile and Maximum values are calculated using Graphpad Prism tool [113].

For RQ2, we analyse and discuss the difference in responses obtained from the two different domains (i.e Software and Agriculture). For this purpose, the data were split into two groups on the basis of user’s demographics information. In the demographic part the questionnaire, users were asked to rate their experience according to their specific domain of experience. The users selected their experience from 1 (no experience of software development) to 5 (Very Experienced software developer) and similarly user from agriculture background are asked to rate their experience from 1 (Never planted anything) to 5 (Very Enthusiastic gardener). Due to the similarities in VI and I, as well as R and VR, answers were aggregated for the purpose of analysis. This approach has also been used in previous studies [121, 122].

6.4 Results and Discussion

This section describes the overall results obtained from survey data.

6.4.1 Number of Respondents:

Figure 6.4.1-1 illustrates the total number of respondents who participated in survey based on their domain (n = 49). The total number of participants from software development domain is 33 and the total number of participants from the agriculture domain is 16. The agriculture group also included those participants who have some experience in software development. There were total of 16 participants in this study who did not have any experience in software development and were mainly from agriculture domain.
6.4.2 Results of RQ1

As mentioned earlier, the main purpose of RQ1 is to evaluate the relevance of the additionally added steps from frame from the users’ perspective. The descriptive statistics of aggregated scale results obtained for each additional step (Q1,..,Q48) in use case are presented in table 6.4.2-1.

Table 6.4.2-1 Descriptive analysis of feedback received

<table>
<thead>
<tr>
<th></th>
<th>VI(1)</th>
<th>I(2)</th>
<th>N(3)</th>
<th>R(4)</th>
<th>VR(5)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Number of feedback</td>
<td>48</td>
<td>48</td>
<td>48</td>
<td>48</td>
<td>48</td>
</tr>
<tr>
<td>Minimum</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>5</td>
<td>18</td>
</tr>
<tr>
<td>25% Percentile</td>
<td>2</td>
<td>3</td>
<td>2</td>
<td>8</td>
<td>25</td>
</tr>
<tr>
<td>Median</td>
<td>3</td>
<td>4</td>
<td>3</td>
<td>11</td>
<td>29</td>
</tr>
<tr>
<td>75% Percentile</td>
<td>3.75</td>
<td>6</td>
<td>4</td>
<td>13</td>
<td>31</td>
</tr>
<tr>
<td>Maximum</td>
<td>5</td>
<td>8</td>
<td>9</td>
<td>17</td>
<td>36</td>
</tr>
<tr>
<td>Sum</td>
<td>133</td>
<td>201</td>
<td>158</td>
<td>504</td>
<td>1347</td>
</tr>
</tbody>
</table>
A total of 48 additional steps were added in different use case of GMS from frames. It is shown in the tables that the values for Minimum, Lower quartile, Median, Upper quartile and Maximum of rate (R and VR) are higher than those of (VI and I). This suggests that majority of the users of these additional 48 steps found them to be relevant. Figure 6.4.2-1 below provides a breakdown of scale ratings given by users on an aggregate basis.

Figure 6.4-2 Percentage of responses rated by all respondents

Figure 6.4.2-1 shows that majority of the participants (57.4%) rated the additional steps very relevant. Overall more than 70% of the participants of the study found the additional steps to be either relevant (21%) or very relevant (57%). In contrast, there were only 15 per cent of the participants who rated the additional steps as either irrelevant (9%) or very irrelevant (6%). Based on this result, it can be concluded that majority of the participants found the additional steps very relevant as a support tool.
6.4.3 Results of RQ2

The second research question of the study is to determine how the responses of the technical and non-technical users rate the additional information?

Figure 6.4.3-1 illustrates the percentage of response rated by software group users. Overall, more than 80% of the participants of the study found the additional steps to be either relevant (22%) or very relevant (60%). In contrast, there were only 13 % of the participants who rated the additional steps either irrelevant (7%) or very irrelevant (6%). Based on this result, it can be concluded that majority of the participants from software group found the additional steps very relevant for GMS.

![Figure 6.4.3 Percentage of responses rated by respondents from software domain](image)

Figure 6.4.3-2 illustrates the percentage of responses rated by agriculture group users. Total 73% of the respondents found the additional steps to be either relevant (20%) or very relevant (53%). In contrast, there were only 15 % of the respondents who rated the additional steps as either irrelevant (11%) or very irrelevant (5%). Based on this result, it can be concluded that majority of the participants from agriculture group found the additional steps very relevant for GMS.
It can be seen in figure 6.4.3-1 and figure 6.4.3-2 that, more than 50 per cent of respondents in both domains (Software development and Agriculture) rated the information added in use case as relevant for all 48 additional steps. Closer inspection does show that the users in the software development domain rated the additional steps higher when compared with agriculture domain users. The reason for this can be that participants from the software development domain have some experience in gardening as well, while the participants from agricultural domain have little or no knowledge or experience of software development.

For further insight, into how the different additional information (Q1,…Q48) is treated by different group, we provide a breakdown of the ‘relevant’ and ‘irrelevant’ ratings for all additional steps, divided according to respondent’s domain (Software and Agriculture).

Figure 6.4.3-3 provides a breakdown of ‘relevant’ rating for all additional steps, divided according to respondent’s domain. In this case, ‘relevant’ covers the range of scale responses covering both VR and R.

The difference in ratings of additional steps between the two groups is not large. For example from (Q1…Q12) there is a slightly low difference in rates of both groups. However from (Q13,…Q24) a larger difference in ratings is evident for steps Q15, Q17, Q20, Q24. The Q15 step is about harvesting and added from Food-Gathering frame, which might not be important for participants from the agricultural domain. Q17 and Q20 are steps of “Create user profile” (sign up and login details). This information does not
actually belong to the agriculture domain and as such it might not be an important step for agriculture domain people. The Q24 “The system fills in the year long timetable with the list of tasks to be carried out for the given plant cultivation” is rated 90.9% and 68.8% by software and agriculture domain participants respectively. The Q24 is a step of “Create Schedule” use case. Since in system overview, it is clearly required to have a timetable to manage and carry out different activities, this could explain the higher relevance of this step for software development participants compared with agricultural domain participants. Similarly we can see difference in rates from (Q25,..Q37) which are the steps of use case “Create Schedule”, “Generate Alert”, “Weather Alert” respectively.

The feedback rated higher by agriculture users than the software people are Q8 (81.3% vs 72.7%), Q18 (68.8% vs 66.7%), Q19 (68.8% vs 66.7%), Q46 (81.3% vs 75.8%). The Q8 is a step of “Create Plant Profile” use case. This step gives information about tools which are required for gardening, thus it might not be much important for software people, but the agriculture domain people know the importance of tools for gardening. The Q18 and Q19 are rated higher in agriculture group but there is not much difference in two groups. These steps are from “Create user profile” use cases where Q18 asks the users to “enter location of signup” and Q19 “purpose for joining the group”. This shows that people from software domain have not considered these as important as by people from agriculture domain to be the part of “Create user profile”. The Q46 is a step of “Weather Update” use case which is added from Perception frame.
Figure 6.4-5 Relevant responses percentage

Figure 6.4.3-4 below provides a breakdown of ‘irrelevant’ rating for all additional steps divided according to respondents. In this case, ‘relevant’ covers the range of scale responses covering both VR and R.

In terms of irrelevant information, the majority of the feedback is rated as irrelevant by the people from agriculture domain as they have rated these less relevant than people from software domain. However, we can see that there are some elements (Q2, Q8, Q11,
Q14, Q15, Q19, Q22, Q41, Q42, Q45, Q46, Q47, Q48) which are rated more irrelevant by software domain people than agriculture.

The Q2, Q8, Q11, Q14 and Q15 are the steps added in “Create Plant Profile” use case by using Agriculture frame. Since this information are purely from gardening contextual, that’s why software people might think it more irrelevant.

The Q19 “purpose for joining the group” is considered more irrelevant by software domain people, as they may not consider this information as important to be the part of “Create User Profile”.

The Q22 “The system records the login time of the user” is a step of “Login” use case, since this step cover none functional requirements, therefore, it might not be considered relevant to the part of use case.

The Q41, Q42, Q45, Q46, Q47 and Q48 are steps of “Weather Update” use case. Since the weather has strong association with agriculture domain, therefore it is considered more relevant by the people from agriculture group.
The result of this study shows that the majority (more than 60%) of users from both domains (Software and Agriculture) considered FrameNet frames as a useful knowledgebase. Further, the elements added in use case from frames is rated equally important by users from both software and agricultural domain. Hence, this gives us a direction that FrameNet frames can help in obtaining a clear set of requirements with a minimal chance of misunderstanding between the analysts and the customers.

Figure 6.4-6 Irrelevant responses percentage
Moreover, there was some information which was rated higher by one group and lower by the other group. For example, the information related to “Sign up” process of GMS, was rated higher by software domain people because this information is more related to technical aspect of system. Similarly, the information related to “Instruments required for gardening” is rated higher by agriculture expertise as this is purely about agriculture domain. Thus, using FrameNet frames for requirements elicitation, allow the technical and non-technical users to come up with a common set of requirements based on their mutual understanding, which can reduce a chance of missing requirements.

6.5 Threats to validity

This section describes the internal and external threat validity of study.

6.5.1 Internal Validity

The internal validity is related to design of the study. The sample size and sample selection make the experiment prone to miss a relevant subgroup of targeted population which decreases the ability to find something really in the population and bias the selection of population. In current study the respondents of survey were asked to nominate and forward the survey link to other people which bias the selection of population. To reduce this threat, it will be more interesting to choose respondents from familiar groups/population in future studies by conducting survey among users belong to software Industries and domain expertise.

6.5.2 External Validity

External validity is used to determine the extent to which the results of a study can be generalized to other situations and to other people. In this study, the users of the study were asked to give their feedback on additional steps added in previously developed use cases. The main threat in the design of the study is about a single domain only i.e. Agriculture domain only. This risk was reduced by the fact that the set of use cases of GMS is not specifically about gardening. There are few use cases which tells about other domain activities such as “Create User Profile”, “Online-Forum”, “Log-In”, “Weather Update”.  

130
6.6 Summary

In this chapter, a quantitative study is conducted to get feedback from participants about additional information added from frames in use cases. The targeted population of the survey was sixty-eight, where forty-nine users participated in study. The aim of the study was to evaluate the support of FrameNet frames in use case elicitation by technical and non-technical people. The survey was conducted among software and agriculture experts. Data is analyzed using descriptive statistics.

These results indicate that the user from different domains considered FrameNet frames as useful knowledgebase for requirements elicitation. Majority of additional information is considered equally important both by software and agriculture domain people, which shows that frames can make easy for requirements analysts to interpret and integrate information gathered from varied communities.
Chapter 7

7 Conclusions and Future work

This chapter describes the conclusion of the research carried out in this thesis. It first provides a summary and conclusions, followed by some directions for further research.

7.1 Thesis Summary

The work contained in this thesis aimed to support requirements analysts in semantics understanding of NL based requirement during requirement elicitation. The main focus was to overcome the problem of conceptual mismatch in NL requirements by understating semantics of text-based requirements in light of linguistic knowledge. The Berkeley FrameNet frame project was used as a mechanism to improve software requirements. FrameNet has the ability to link concepts to each other through frames and FEs. This can prompt requirements engineers to consider ancillary concepts that might otherwise have been omitted.

The proposed approach has been illustrated by adopting a mixed methodology [117] and demonstrated that how FrameNet frames can support use case elicitations. The approach uses frames to provide suggestions, where the FEs show actors, how the actions should be organised, what roles should be fulfilled for the coherence of a given concept, and what sequence of interactions could be expected.

A set of predefined metrics are used to evaluate the quality of use case elicit with frames. The role of frames as domain knowledge is also evaluated by conducting survey among domain expert people. It is also investigated that how the different users used frame for use case elicitation and work on identifying common patterns they adopted to elicit use cases.

Chapter 2 provided background information of requirements elicitation and the problems faced by analyst using NL for requirements elicitation. This chapter also provided the details of techniques and approaches used for solving issues of NL. A detail of use case, use case quality and the metrics used to measure the use case quality has also been given. The detailed overview of FramNet is also explained in this chapter. Finally, the guide lines and approaches, previously adopted to obtain good quality use cases are discussed.
In Chapter 3 the proposed methodology is implemented on two subject systems to get preliminary result about our hypothesis. The initial results indicated that the use of FrameNet frames for use case elicitation augmented semantic understanding and provide insight to further elaborate the frame components for identification of missing and related requirements.

In Chapter 4, an experimental study is carried out to explore the difference in the quality of use cases produced with and without frame. The research question in this study was “RQ1: How helpful the FrameNet for obtaining Clear, Complete and Consistent use cases?” The study was conducted among novel software engineering students where the participants of the study were divided into control and experimental groups. The participants were given task to develop a set of use cases with and without frame. A set of metrics proposed by Torner et al. [26] was used to evaluate the consistency, completeness, and clarity of use cases. The outcomes showed that the FrameNet frames may produce a vital improvement in quality of a use case, and clearer, complete and consistence use cases are produced with frames.

Chapter 5 investigated two research questions about requirements elicitation using frames. The typical research questions were “RQ1: How helpful is FrameNet for creating use cases?” and “RQ2: How is the FrameNet used by requirement analysts in creating the use cases? In this regard, a qualitative study is conducted among professional requirements analysts to determine how frames are utilized in use cases elicitation in terms of valid and invalid information. The result of RQ1 indicated that the use of frame is supportive to elicit the valid and relevant requirements. The results of the study have shown that the FrameNet supports frame specialization to reflect the patterns of language use. We identified some of the common patterns adopted by different participant using frame for us case elicitation. However, the frames are not always the same for all speaker as frames are based on individual experience, thus we cannot finalize fully common patterns using frames for use cases elicitation.

The chapter 6 investigated the usefulness of frame on a large scale of experienced users. The exact research questions were “RQ1: How relevant/useful the additional steps are added from frames for a given system?” and “RQ 2: Is the additional information provided by FrameNet equally useful by participants having experience in gardening and software development?”. In this regard, a survey based study is conducted among software developer and domain expert to gather their feedback about information added
in use cases with frames. The study results, complement the use of frames for use cases both by software developer and domain expert. It gives a direction that FrameNet frames can help in obtaining a clear set of requirements with a minimal chance of misunderstanding between the analysts and customers.

7.2 Conclusion

In this thesis, a novel approach is proposed to elicit use cases using lexical database FrameNet Frames. The investigation indicated that FrameNet frames are a valuable prompt for enhancing the semantic understanding of text based requirements, as well as for identification of missing and related requirements. The research findings have supported the idea that the use of frames in use case-based elicitation leads to a clearer understanding of the domain concepts covered in a use-case and to help acquire complete information. This research provides evidence to construct clear, complete and consistent use cases by utilizing different components of FrameNet frames.

The research findings confirm that the construction and understating of knowledge can be better done with FrameNet frames, because in FrameNet frames the different activities, objects and people are organized (represented by different FEs) in sequences to develop a semantic understanding of a particular concept. The proposed approach has evaluated, using qualitative and quantitative studies. The results indicate that frames lead to clearer understanding of the domain concepts covered in a use-case and acquired complete information.

7.3 Future work:

The work conducted in this thesis has met the objectives set for this research and has made significant contributions in the area of requirements elicitation. This work can be extended in various dimensions in the future. Some of the main directions for future work are defined as follows:

Extension of the experimental work

The experiments and case studies presented in this thesis provide useful results and contributions to use case elicitation. However, there is scope for further extending the experimentations by taking into account quantitative studies in different domains, to generalize FrameNet’s performance.
Reusability of frames

During experiments we come across few frames which are used in different domains. For example, the frame “Membership” is used for “Log In” use case in GMS and for “Register new employee” use in HWS (the detail of use cases is available in appendix A). Thus, there is need to explore how efficient the frames are requirements usability.

Modularity

This research provided some evidence to suggest that the frames are helpful in identifying interrelated requirements as described in chapter 3. The linguistics frames can provide an excellent starting point for requirements modularisation. We consider our notion of frame-concerns to be closely related to the work on Multidimensional Separation of Concerns (MDSOC) [123]. Similar to MDSOC slices, frames focus on requirements level concern at a time. Each of these frame-concerns is an independent fully formed view on a particular (defined) problem. Each frame can have FEs that overlap or repeated in other frames, and this overlap is the basis of frame composition and larger-scale meaning building. Hence, it is suggested that the frame-concerns serve as an explanation for the origin of hyperslices on basis of NL motivated schematization of human experience. Further studies to validate this will be undertaken by tracing the notions modelled in MDSOC to the language level frames.
Appendix

Appendix A

A-1. Garden Management System Overview:

“The idea is that software will function as interactive garden manager. It will provide a year-long schedule on a monthly by monthly basis. Users will need a log-in to access their individual profiles in order to tailor the garden manager to their needs. Users should then be able to select which plants they intend to grow, and whether or not each of them is indoor, outdoor, or a ‘porta plot’ gardening. Once a user has created their garden profile the software should then inform them of what needs to be done in the garden possibly in the form of alerts or reminders. This should cover times to harvest, times to sow or plant, and general maintenance around the garden. Ideally this will also include tips or ‘how-to’s’, for instance, instructions on how to tell when the fruits are ready to be picked, or how to dry herbs, ideal spacing for sowing, how to prune, how to tell if something needs watering and so on. Additional elements of the software could include weather updates, recommendations for companion planting (what grows well next to what), recommendations for inter-planting (what can be grown with what, e.g. beans up sweet corn stalks), pest watch, and data collection to provide a year on year comparison for gardeners and recommended rotations. To introduce more complex security there is also the potential for the software to have a forum function, so that gardeners may share advices and experiences, without other users being able to access their personal information”
A-2. Health Watcher System Overview:

The purpose of the system is to collect then manage public health related complaints and notifications. The system is also used to notify people about important information regarding the Health System. The Health Watcher System must also exchange information with the SSVS system (Sanitary Surveillance System). Initially, this exchange will involve the querying of sanitary licenses. Subsequently, when the SSVS has the Complaint Control module deployed, Sanitary Surveillance complaints will be exchanged between the two systems.

Broadening and related systems:

With the deployment of Health-Watcher System, the Public Health System will considerably improve:

- The complaint control (registering and notifications).
- Quality of service regarding the dissemination of information; for example:

  Vaccination campaigns, disease prevention, health guides, obtaining birth/death certificates and application details for a sanitary license. The system will be managed by DIEVS and will exchange information with the Sanitary Surveillance system.

A citizen can access the system through the internet or dialing 1520, and make their complaint or ask information about the health services. In the event of a complaint being made, it will be registered on the system and addressed by a specific department. This department will be able to handle the complaint in an appropriate manner and return a response when complaint has been dealt with. This response will be registered on the system and available to be queried. The system will be for public use in kiosks at several strategic points, on which the citizen will be able to register complaints and request information.
Appendix B

B-1. Experiment Introduction

Dear 2nd year students,

I am a PhD student at the Department of Informatics University of Leicester, and I am welcoming you to participate in a short study to help evaluate my research idea.

This will take about 1 hour and 15 minutes of your time, here are some details about the study:

The study asks you to write a three use cases with and without some additional reference materials. In order to help remind you what use cases are, I will give a short (10 min.) refresher talk. After this, each participant will be asked to:

Firstly you will be given a system overview of restaurant management system and will be asked to write use cases for a few user interactions with the intended software system. During the exercise, write the use case in textual format on the sheets of paper provided. If there are any parts of this interaction that you cannot recall or are not familiar with, please make assumptions.

Some participants will be provided with a set of pre-prepared models known as frame. Please use these models as you see fit to ensure the use case is correct, complete, consistence and clear.

The point of this task is to see how these pre-prepared models helps to obtain more complete, consistence and clear use cases. This will take approximately 45 min.

B-2. Restaurants System overview

In a popular London restaurant, the following system is required to speed up preparation of meals. Each waiter is assigned a group of tables, after taking orders for a table the waiters enter the orders (a list of dishes and drinks ordered by the diner or group of diners) in the system at the PC. The waiter usually knows about any item that is not available before taking an order but occasionally one of the special items would have already been sold out before taking the order. The system must confirm the availability of dishes. Should an item not be available the system must allow the waiter to change or even delete a customer’s order. Food items to be prepared are sent to the kitchen, drinks orders to the
bar. Starters and main course orders are usually taken together. Drinks and desert orders may be taken separately.

Kitchen staff sees the dish orders on their screen, prepare them in an appropriate sequence and confirm preparation to the system when complete, similarly with the bar.

When a waiter sees the completion indications on his terminal he collects the items and takes them to the table. The waiter can also check on the status of dish and drink orders.

At the end of the meal the waiter will have the system print a bill, and he will enter the details of payment for it. The management can give discounts.

The system keeps track of the numbers of customers served by each waiter and the amount of money taken by each waiter. The management can view these statistics.

**B-3. Consent Form**

Title of the study: -----------------------------

Name of researcher:---------------------------------

Name of the participant:----------------------, Department:------------------, Email:-------------

---

a.  I confirm that I have read and understand the information for the study and have had the opportunity to ask questions.

b.  I understand that my participation is voluntary and that I
    a. am free to withdraw at any time, without giving reason.

c.  I agree to take part in the above study.

d.

e.  I agree to the interview / focus group / consultation being audio recorded

f.  I agree to the interview / focus group / consultation being video recorded

---
B-4. Selected Frames

B-4.1. “Request Entity” frame is selected for “Input Order” Use case

Request_Entity definition:

A Customer requests an Entity from a Supplier. The Customer can order the Entity himself or via a Medium.

For example:

I ORDERED a black scarf from J. Crew a week ago.

Mr. Strom's ORDER has been cancelled.

Core FEs:

Customer [Cust]  The Customer is the person or business ordering the Entity from the Supplier.

Simone ORDERED a hamburger and fries.

Entity [Ent]  The object ordered by the Customer.

Liz ORDERED a new espresso maker for the office.

His ORDER did not arrive on time.
Supplier [S] The person or organization from whom the Customer orders the Entity.

Mrs. Apfelbaum ORDERED a coffee from the barista.

Non-Core FEs:

Beneficiary [Bene] The person or organization which benefits from the Entity.

Rose ORDERED a new computer for Mr. Cooper.

Manner [Man] The Manner in which the request is placed.

Mr. Shaw hastily ORDERED the first thing he saw on the menu.

Means [Means] An action performed by the Customer that results in a request being made.

Cheri accidently ORDERED the jacket by clicking on the link.

Medium [Med] The channel used by the Customer to transmit the request.

Harold ORDERED the material via email.

Time [Time] The Time at which the Customer requested the Entity.

Sara placed the ORDER three days ago.
B-4.2. “Billing” frame is selected for “Print Bill” Use case

Billing definition:

“A Billing_authority communicates in written form that the Debtor owes the Owed_party (often the same as the Billing_authority) some amount of Money for certain Goods (including services) provided by the Owed_party or to clear a debt for a Bad_action by the Debtor against the Owed_party.”

Core FEs:

Billing_authority The individual who communicates to the Debtor the Money that they owe.

Debtor [Byr] The person (or organization) that owes the Owed_party.  

The pizza man BILLED Lee $35 for his order.

Goods [Gds] The FE Goods is anything (including labor or time, for example) which is exchanged for Money in a transaction.

The man at the counter BILLED Lee for his dry-cleaning.

Money [Mny] Money is the thing given in exchange for Goods in a transaction, or as recompense for the Bad_action.

The clerk INVOICED the little old lady $49.95 for what she called "a little hand-holding."

Owed_party [Slr] The Owed_party expects a specific amount of Money from the Debtor, either because the Debtor received Goods or because the Debtor performed a Bad_action against the Owed_party.

The salesman never BILLED me, so I did not pay.

Rate [Rate] In some cases, price or payment is described per unit of Goods.

The cleaner BILLS at forty dollars per hour for his services.
**Non-Core FEs:**

<table>
<thead>
<tr>
<th>Explanation</th>
<th>The Explanation for which an event occurs.</th>
</tr>
</thead>
<tbody>
<tr>
<td>[Exp]</td>
<td></td>
</tr>
<tr>
<td>Semantic</td>
<td></td>
</tr>
<tr>
<td>Manner</td>
<td>[] Any description of the billing event which is not covered by more specific FEs, including secondary effects (quietly, loudly), and general descriptions comparing events (the same way). It may also indicate salient characteristics of the Seller that affect the action (presumptuously, coldly, deliberately, eagerly, carefully). He carefully COLLECTED all his debts.</td>
</tr>
<tr>
<td>Means</td>
<td>[Mns] The means by which the billing occurs. The money was BILLED through a credit card transaction.</td>
</tr>
<tr>
<td>Place</td>
<td>[Place] Where the event takes place.</td>
</tr>
<tr>
<td>Purpose</td>
<td>[Purp] The purpose for which an intentional act is performed.</td>
</tr>
<tr>
<td>Time</td>
<td>[Time] When the event occurs.</td>
</tr>
<tr>
<td>Unit</td>
<td>[Unit] This FE is any unit in which goods or services can be measured. Generally, it occurs in a by or per-PP. I need a per-item INVOICE!</td>
</tr>
</tbody>
</table>
B-4.3. “Commerce_Pay” frame is selected for “Input payment details” Use case

Commerce_pay definition:

This frame involves Buyers paying Money for Goods. In this frame the Money is the direct object, and is mapped to the theme of the transfer.

For example:

I PAID her $50 for a video game.

Core FEs:

<table>
<thead>
<tr>
<th>Role</th>
<th>Description</th>
</tr>
</thead>
</table>
| Buyer [Byr] | The Buyer has the Money and wants the Goods. 
| Gem PAID fifty dollars for a shirt |
| Goods [Gds] | The FE Goods is anything (including labor or time, for example) which is exchanged for Money in a transaction. 
| Kim PAID fifty dollars for the shirt |
| Money [Mny] | Money is the thing given in exchange for Goods in a transaction. 
| Pat PAID 14 dollars for a movie ticket. |
| Rate [Rate] | In some cases, price or payment is described per unit of Goods. 
| The manager PAYS the paper boys five dollars an hour. |
| Seller [Slr] | The Seller has the Goods and wants the Money. 
| The landlord is PAID $700 a month by the tenants for the apartment. |

Non Core FEs:
Circumstances [Cir] | Circumstances describe the state of the world (at a particular time and place) which is specifically independent of the event itself and any of its participants.

Worker’s compensation PAYMENTS alone in cases involving falls, now average more than $8500 per incident in North America.

Explanation [Exp] | The Explanation for which an event occurs.

Frequency [Fre] | This frame element is defined as the number of times an event occurs per some unit of time.

A good tenant PAYS her rent regularly

Manner | Any description of the paying event which is not covered by more specific FEs, including secondary effects (quietly, loudly), and general descriptions comparing events (the same way). It may also indicate salient characteristics of the Buyer that affect the action (presumptuously, coldly, deliberately, eagerly, carefully).

She quietly PAID the bill and left.


Abby BOUGHT the car with cash. Robin PAID for the car by check.

Place [Place] | Where the event takes place.

Purpose [Purp] | The Purpose for which an intentional act is performed.

Time [Time] | When the event occurs.
This FE is any unit in which goods or services can be measured. Generally, it occurs in a by-

Bob PAYS for peppers by the pound.
Appendix C

C-1. Email:

What does the study involve?

The study asks you to write few use cases with and without some additional reference material. In order to help you remind what use cases are, I will give a short (10 min) refresher talk. The participants will be given chance to ask question about study before the interview takes place.

After this, each participant will be asked to:

1. Write use cases for few user interactions with the intended software system. Some participants will be asked to refer to a set of pre-prepared models, while others will be asked to write use cases without any additional models. The point of this task is to see how these pre-prepared models change/help/hinder writing of the use cases. This will take approximately 40 min.

2. During this interview, I will ask you to explain how you write steps and from where you get particular idea to write specific steps of use case. The interview will be scheduled after one week.

Please note that both parts of the study will be audio recorded in order to collect materials for analysis of the use case creation and model use process.

C-2. Demographic

The information sheet provided by participants helped us to equally spread experience and qualification amongst both groups.

<table>
<thead>
<tr>
<th>S.No.</th>
<th>Gender</th>
<th>Years of experience in use cases</th>
<th>Type of experience</th>
<th>Current Status</th>
<th>Educational Background</th>
<th>Location</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Male</td>
<td>3-5 Years</td>
<td>Industrial related</td>
<td>Senior Quality Engineer</td>
<td>Master computer science</td>
<td>Pakistan</td>
</tr>
<tr>
<td>2</td>
<td>Male</td>
<td>5-10 Years</td>
<td>Industrial related</td>
<td>IT Business Analyst</td>
<td>Master computer science</td>
<td>USA</td>
</tr>
<tr>
<td>No.</td>
<td>Gender</td>
<td>Experience</td>
<td>Interests</td>
<td>Qualification</td>
<td>Country</td>
<td></td>
</tr>
<tr>
<td>-----</td>
<td>--------</td>
<td>------------</td>
<td>----------</td>
<td>---------------</td>
<td>---------</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>Female</td>
<td>1-3 Years</td>
<td>Study and Teaching related</td>
<td>Lecturer</td>
<td>PhD computer science</td>
<td>UK</td>
</tr>
<tr>
<td>4</td>
<td>Male</td>
<td>1-3 Years</td>
<td>Study related</td>
<td>PhD Student</td>
<td>Master computer science</td>
<td>Portugal</td>
</tr>
<tr>
<td>5</td>
<td>Male</td>
<td>3-5 Years</td>
<td>Study and Teaching related</td>
<td>PhD Student</td>
<td>Master computer science</td>
<td>UK</td>
</tr>
<tr>
<td>6</td>
<td>Male</td>
<td>3-5 Years</td>
<td>Industrial related</td>
<td>Senior Software Developer</td>
<td>Bachelor in computer science</td>
<td>Pakistan</td>
</tr>
<tr>
<td>7</td>
<td>Male</td>
<td>5-10 Years</td>
<td>Industrial related</td>
<td>Advisory Quality Engineer</td>
<td>Bachelor in computer science</td>
<td>Pakistan</td>
</tr>
<tr>
<td>8</td>
<td>Male</td>
<td>3-5 Years</td>
<td>Teaching and Industrial related</td>
<td>Lecturer</td>
<td>PhD in computer science</td>
<td>Pakistan</td>
</tr>
<tr>
<td>9</td>
<td>Female</td>
<td>0-1 Years</td>
<td>Study related</td>
<td>PhD Student</td>
<td>Master in computer science</td>
<td>UK</td>
</tr>
<tr>
<td>10</td>
<td>Male</td>
<td>3-5 Years</td>
<td>Teaching, Study and Industrial related</td>
<td>PhD Student</td>
<td>Master in computer science</td>
<td>Hong Kong</td>
</tr>
</tbody>
</table>

C-3. Restaurants System Overview:

“In a popular London restaurant, the following system is required to speed up preparation of meals. Each waiter is assigned a group of tables, after taking orders for a table the waiters enter the orders (a list of dishes and drinks ordered by the diner or group of diners)
in to the system at the PC. The waiter usually knows of any dishes that are
 unavailable before taking an order but occasionally one of the specials will sell out.
The system must confirm the availability of dishes. Should an item not be available
the system must allow the waiter to change or even delete a customer’s order. Dishes
to be prepared are sent to the kitchen, drinks orders to the bar. Starters and main
course orders are usually taken together. Drinks and desert orders may be taken
separately.
Kitchen staff sees the dish orders on their screen, prepare them in an appropriate
sequence and confirm preparation to the system when complete, similarly with the
bar.
When a waiter sees the completion indications on his terminal he collects the items
and takes them to the table. The waiter can also check on the status of dish and
drink orders.
At the end of the meal the waiter will have the system print a bill, and he will enter the
details of payment for it. The management can give discounts.
The system keeps track of the numbers of customers served by each waiter and the
amount of money taken by each waiter. The management can view these statistics.”

C-4. Selected Use Cases

Use case ID: 1

Use case: Input order

Actors: ?

Goal: To input an order for a meal

Description: When a Diner orders a meal, the waiter writes down the order and puts it
into the system. The system presents this order to the Kitchen staff who prepares the food.

Pre-Condition: ?

Post-Condition: ?
Normal Events:?
Alternative Events: ?

Use case ID:2
Use case: Input payment details

Actors:

Goal: To input payment details to the system

Description: The Waiter takes the payment from the Diner and uses the system to input the details about the payment.

Pre-Condition: ?
Post-Condition: ?
Normal Events:?
Alternative Events: ?

C-5. Selected Frames

“Request Entity” frame is selected for “Input Order” Use case

[Diagram showing the flow of information between Customer, Entity, and Supplier]
visual representation of Request-Entity Frame

C-5.1. Request_Entity

A Customer requests an Entity from a Supplier. The Customer can order the Entity himself or via a Medium.

For example:
I ORDERED a black scarf from J. Crew a week ago.

Mr. Strom's ORDER has been cancelled.

FEs:
Core:

Customer [Cust] The Customer is the person or business ordering the Entity from the Supplier.

Simone ORDERED a hamburger and fries.

Entity [Ent] The object ordered by the Customer.

Liz ORDERED a new espresso maker for the office.

His ORDER did not arrive on time.

Supplier [S] The person or organization from whom the Customer orders the Entity.

Mrs. Apfelbaum ORDERED a coffee from the barista.

Non-Core:

Beneficiary [Bene] The person or organization which benefits from the Entity.

Rose ORDERED a new computer for Mr. Cooper.

Manner [Man] The Manner in which the request is placed.

Mr. Shaw hastily ORDERED the first thing he saw on the menu.
Means [Means] An action performed by the Customer that results in a request being made.

Cheri accidently ORDERED the jacket by clicking on the link.

Medium [Med] The channel used by the Customer to transmit the request.

Harold ORDERED the material via email.

Time [Time] The Time at which the Customer requested the Entity.

Sara placed the ORDER three days ago.

“Commerce_Pay” frame is selected for “Input payment details” Use case

**Visual representation of Commerce_Pay Frame**

*C-5.2. Commerce_Pay*

This frame involves Buyers paying Money for Goods. In this frame the Money is the direct object, and is mapped to the theme of the transfer.
For example:

I PAID her $50 for a video game.

FEs:
Core:

Buyer [Byr]  The Buyer has the Money and wants the Goods.

Gem PAID fifty dollars for a shirt

Goods [Gds]  The FE Goods is anything (including labor or time, for example) which is exchanged for Money in a transaction.

Kim PAID fifty dollars for the shirt

Money [Mny]  Money is the thing given in exchange for Goods in a transaction.

Pat PAID 14 dollars for a movie ticket.

Rate [Rate]  In some cases, price or payment is described per unit of Goods.

The manager PAYS the paper boys five dollars an hour.

Seller [Slr]  The Seller has the Goods and wants the Money.

The landlord is PAID $700 a month by the tenants for the apartment.

Non-Core:

Circumstances [Cir]  Circumstances describe the state of the world (at a particular time and place) which is specifically independent of the event itself and any of its participants.

Worker's compensation PAYMENTS alone in cases involving falls, now average more than $8500 per incident in North America.

Explanation [Exp]  The Explanation for which an event occurs.

Frequency [Fre]  This frame element is defined as the number of times an event occurs per some unit of time.

A good tenant PAYS her rent regularly
Manner

[] Any description of the paying event which is not covered by more specific FEs, including secondary effects (quietly, loudly), and general descriptions comparing events (the same way). It may also indicate salient characteristics of the Buyer that affect the action (presumptuously, coldly, deliberately, eagerly and carefully).

She quietly PAID the bill and left.

Means

[Mns] The Means by which a commercial transaction occurs.

Abby BOUGHT the car with cash. Robin PAID for the car by check.

Place

[Place] Where the event takes place.

Purpose

[Purp] The Purpose for which an intentional act is performed.

Time

[Time] When the event occurs.

Unit [Unit]

This FE is any unit in which goods or services can be measured.

Generally, it occurs in a by-PP

Bob PAYS for peppers by the pound.
C-6. Results

C-6.1. Data of Research Question 2

Schematic representation of participant 2
Schematic representation of participant 3
Schematic representation of participant 4
Schematic representation of participant 5
## D-1. Survey Questioner

<table>
<thead>
<tr>
<th>Questions</th>
<th>Use Cases</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Q1</td>
<td>Create Plant Profile</td>
<td>[2. System asks about the place (geographical location) of user where he/she intends to grow the plant.]</td>
</tr>
<tr>
<td>Q2</td>
<td></td>
<td>[3. User provides the information about his/her location.]</td>
</tr>
<tr>
<td>Q3</td>
<td></td>
<td>[4. System displays the list of plants can grow at that particular place.]</td>
</tr>
<tr>
<td>Q4</td>
<td></td>
<td>[4.1. The system highlights a list of plants available for the given location]</td>
</tr>
<tr>
<td>Q5</td>
<td></td>
<td>[8.1 Where to grow the plant (physical location, e.g., indoor, outdoor)]</td>
</tr>
<tr>
<td>Q6</td>
<td></td>
<td>[8.2 What time to grow the plant (i.e., month to sow)]</td>
</tr>
<tr>
<td>Q7</td>
<td></td>
<td>[8.3 Ideal spacing between plants for sowing.]</td>
</tr>
<tr>
<td>Q8</td>
<td></td>
<td>[8.4 The tools need to sow/grow the plant.]</td>
</tr>
<tr>
<td>Q9</td>
<td></td>
<td>[8.5 How often the user can grow this plant]</td>
</tr>
<tr>
<td>Q10</td>
<td></td>
<td>[8.6 The starting state of a plant i.e. seed/plant.]</td>
</tr>
<tr>
<td>Q11</td>
<td></td>
<td>[8.7 The outcome of the plant]</td>
</tr>
<tr>
<td>Q12</td>
<td></td>
<td>[8.8 The duration plant needs to grow]</td>
</tr>
<tr>
<td>Q13</td>
<td></td>
<td>[8.9 Particular iteration require by a plant to grow]</td>
</tr>
<tr>
<td>Q14</td>
<td></td>
<td>[8.10 The amount of effort put into the food gathering event]</td>
</tr>
<tr>
<td>Q15</td>
<td></td>
<td>[8.11 How often the food can be gathered.]</td>
</tr>
<tr>
<td>Q16</td>
<td></td>
<td>[8.12 Source of food gathering]</td>
</tr>
<tr>
<td>Q17</td>
<td></td>
<td>[1. The user clicks on signup option of the system.]</td>
</tr>
<tr>
<td>Q18</td>
<td>Create user Profile</td>
<td>[2.5 Location]</td>
</tr>
<tr>
<td>-------</td>
<td>---------------------</td>
<td>----------------</td>
</tr>
<tr>
<td>Q19</td>
<td>Log In</td>
<td>[2.6 Purpose for joining the group]</td>
</tr>
<tr>
<td>Q20</td>
<td>[5. System records the time when the user becomes a member]</td>
<td></td>
</tr>
<tr>
<td>Q21</td>
<td>Log In</td>
<td>[6. The system checks the user status in the group and sets access privileges accordingly.]</td>
</tr>
<tr>
<td>Q22</td>
<td>[7. The system records the login time of the user.]</td>
<td></td>
</tr>
<tr>
<td>Q23</td>
<td>Online Forum</td>
<td>[5. The system observes discussion duration on each topic and updates the user on past discussions.]</td>
</tr>
<tr>
<td>Q24</td>
<td>Create Schedule</td>
<td>[1. The system fills in the yearlong timetable with the list of tasks to be carried out for the given plant cultivation.]</td>
</tr>
<tr>
<td>Q25</td>
<td></td>
<td>[2. The system adds these tasks to the user’s calendar on monthly biases as follows:]</td>
</tr>
<tr>
<td>Q26</td>
<td></td>
<td>[2.1 When to give water]</td>
</tr>
<tr>
<td>Q27</td>
<td></td>
<td>[2.2 When to pluck the fruit/flower]</td>
</tr>
<tr>
<td>Q28</td>
<td></td>
<td>[2.3 When to watch for pests]</td>
</tr>
<tr>
<td>Q29</td>
<td></td>
<td>[2.4 The system displays a message on successfully created timetable for the plant cultivation to the user.]</td>
</tr>
<tr>
<td>Q30</td>
<td></td>
<td>[2.5 When to sow the plant]</td>
</tr>
<tr>
<td>Q31</td>
<td>Generate Alert</td>
<td>[2.2 Alerts can be in form of an SMS message, email or alarm.]</td>
</tr>
<tr>
<td>Q32</td>
<td></td>
<td>[2.3 The alert can be repeated after a specified period of time, unless it is dismissed.]</td>
</tr>
<tr>
<td>Q33</td>
<td>Weather Alert</td>
<td>[1. System receives weather forecast on an unusual weather event from the weather forecast system]</td>
</tr>
<tr>
<td>Q34</td>
<td></td>
<td>[2. The system generates and sends warning alerts about the unusual weather event to the user:]</td>
</tr>
<tr>
<td>Q35</td>
<td></td>
<td>[2.1 A warning alert is generated ahead of the expected unusual weather]</td>
</tr>
<tr>
<td>Q36</td>
<td>[2.2 A warning alert is sent to the user as and SMS, email, or alarm.]</td>
<td></td>
</tr>
<tr>
<td>-----------</td>
<td>------------------------------------------------------------------------</td>
<td></td>
</tr>
<tr>
<td>Q37</td>
<td>[2.3 The warning is first sent 3 days ahead of the unusual weather, then repeated one day before.]</td>
<td></td>
</tr>
<tr>
<td>Q38</td>
<td>[3. User receives and dismisses the warning alert.]</td>
<td></td>
</tr>
<tr>
<td>Q39 Weather Update</td>
<td>[1. Define the location where user wants to check the weather.]</td>
<td></td>
</tr>
<tr>
<td>Q40</td>
<td>[2. Choose the time frame for the weather forecast (i.e. the time for what the user wants to check weather).]</td>
<td></td>
</tr>
<tr>
<td>Q41</td>
<td>[3. The user provides requested details.]</td>
<td></td>
</tr>
<tr>
<td>Q42</td>
<td>[4. System validates the details and displays information about:]</td>
<td></td>
</tr>
<tr>
<td>Q43</td>
<td>[4.1 Temperature]</td>
<td></td>
</tr>
<tr>
<td>Q44</td>
<td>[4.2 Weather conditions]</td>
<td></td>
</tr>
<tr>
<td>Q45</td>
<td>[4.3 Amount of precipitation]</td>
<td></td>
</tr>
<tr>
<td>Q46</td>
<td>[4.4 Quantity of precipitation that falls within certain duration.]</td>
<td></td>
</tr>
<tr>
<td>Q47</td>
<td>[4.5 Number of times an event occurs per some unit of time]</td>
<td></td>
</tr>
<tr>
<td>Q48</td>
<td>[4.6 The manner of precipitation]</td>
<td></td>
</tr>
</tbody>
</table>
References


83. Fockel, M. and J. Holtmann. A requirements engineering methodology combining models and controlled natural language. in Model-Driven Requirements Engineering Workshop (MoDRE), 2014 IEEE 4th International. 2014. IEEE.
85. Giganto, R. Generating class models through controlled requirements. in NZCSRSC-08, New Zealand Computer Science Research Student Conference Christchurch, New Zealand. 2008.
98. Alistair, C., Writing effective use cases. 2001, Addison-Wesley Reading.
102. kundi, M. Appendix A. 2018 [cited 2018; Available from: https://drive.google.com/uc?export=download&id=1ZXlRsQnQpTgJ_895ip6ANXpQ5gDZFO4m.


