Knowledge Enhanced Experience Reuse for Requirements Engineering Decision Support (KEEREDECS)

A THESIS SUBMITTED

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR THE

MS IN SOFTWARE ENGINEERING

BY

Ikhlaq Ahmed

SUPERVISED BY

DR. AASIA KHANUM

COLLEGE OF ELECTRICAL AND MECHANICAL ENGINEERING,

NATIONAL UNIVERSITY OF SCIENCES AND TECHNOLOGY (NUST)

RAWALPINDI

June, 2014

Department of Computer Engineering

NUST College of E&ME, Rawalpindi

Session (2010-2014)

A thesis submitted to the Computer Engineering Department, College of Electrical and Mechanical Engineering, National University of Sciences and Technology, Rawalpindi in partial fulfillment of the requirements for the degree of Masters of Science in Computer Software Engineering

Approved By

Supervisor: Dr. Aasia Khanum

GC Members

1. Col. Dr. FarooqueAzam
2. Dr. Saad Rehman
3. Dr. Arslan Shaukat

Date:

Declaration

I certify that this research work titled "*Knowledge Enhanced Experience Reuse for Requirements Engineering Decision Support (KEEREDECS)*" is my own work. The work has not been presented elsewhere for assessment. The material that has been used from other sources it has been properly acknowledged / referred.

Signature of Student

Ikhlaq Ahmed

2010-NUST-MS PhD-CSE (E)-03

Language Correctness Certificate

This thesis has been read by an English expert and is free of typing, syntax, semantic, grammatical and spelling mistakes. Thesis is also according to the format given by the university.

Signature of Student

Ikhlaq Ahmed

2010-NUST-MS PhD-CSE (E)-03

Signature of Supervisor

Copyright Statement

Acknowledgments

First I would like to thank Almighty Allah to give me the opportunity and strength to complete my thesis successfully. I would like to thank to my family for their continuous patience, understanding and emotional support during these years.

I would like to express my most sincere appreciation to my supervisor Dr. Aasia Khanum, for her continuous support, intellectual guidance and critical remarks during the whole period of my research.

To my HOD and all of my GC Members, I wish to express my heartiest gratitude to Brig. Dr. Shoab Ahmed Khan,Col. Dr. Farooque Azam,Dr. Arslan Shaukat, and Dr. Saad Rehman on their invaluable advice and encouraging support. They not only supervised me to materialize the work but also showed me the way to think in different dimensions and work steadily from the very beginning of a research. Their continuous support helped me to feel my confinement with the topic all through my research.

At the end I would like to acknowledge the survey participants specially Mr. Sohail Qadeer Abbasi who spent his time and shared his views and experiences.

Dedicated to my exceptional parents and adored siblings whose tremendous support and cooperation led me to this wonderful accomplishment

Abstract

Requirements insufficiency is one of the major contributing factors to software projects failure, indicating that Requirement Engineering (RE) process must be applied judiciously to improve requirements quality. Potentially, many activities can be performed during RE process, and each activity can be supported by several techniques. There is general consensus that RE activities and techniques should be customized to suit contextual features like project, process, product etc. Moreover, a holistic view of the process is needed while deciding the RE activities and techniques. Providing automatic support for fulfilling this need is a challenging problem due to non-deterministic and human-oriented nature of decision-making in the domain. The KEEREDECS framework proposed in this work is an automated approach that will makephase wise contributions to RE process. First, a context processing module is used to get project and techniques context parameters values and weights.Second,Fuzzy Case Based Reasoning (FCBR) technique from Artificial Intelligence is used to handle non-deterministic and context-sensitive decision-making with the help of experiential learning. Third, technique set generations module is used to involve the expert judgment to support decision making.The system can run autonomously or with expert involvement. Evaluation ofKEEREDECS indicates that the approach has a good potential to support for informed decision-making leading to better quality of obtained requirements.

Keywords: *Software Requirements Engineering, Decision Support System, Context Aware Techniques, Fuzzy Logic, Case Based Reasoning, Planning Systems*

Table of Contents

List of Figures

List of Tables

List of Abbreviations

AI	Artificial Intelligence
RE	Requirements Engineering
DECS	Decision Support
CBR	Case Based Reasoning
V&V	Validation and Verification
SE	Software Engineering
KBS	Knowledge Based Systems
FL	Fuzzy Logic
CI	Computational Intelligence
EF	Experience Factory

CHAPTER

1. Introduction

1.1. Software Requirements Engineering

The success or failure of any software product like an e-banking system, an embedded system, or a business intelligence system dependson its acceptability with the various stakeholders,making it imperative that the voices of diversestakeholders are listened to, recorded, collated, andincorporated into the software product [4], [5], [6]. Ithas been established empirically that one of the majorreasons for software project failure is the existence ofunclear, imprecise, or ambiguous requirements [9], [10].

Requirements Engineering (RE) is the branch of systems engineering concerned with the desired properties and constraints of software-intensive systems, the goals to be achieved in the software's environment, and assumptions about the environment [60]. RE is a pivotalprocess in software engineering, which is a basic constituent of systems engineering (Figure 1 [61]).

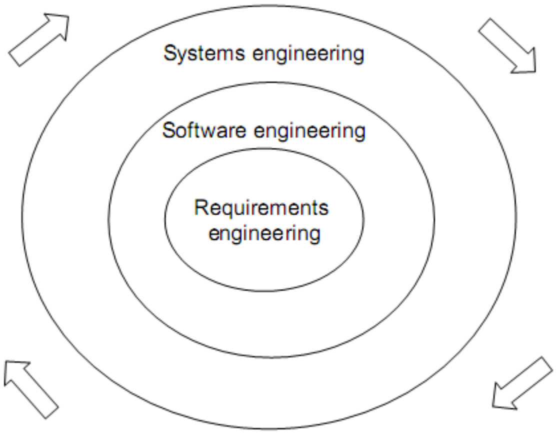


Figure 1. RE Process in Context of Systems Development

RE process involves severalactivities. Figure 2 [61] shows the generic RE process activities involving elicitation, analysis, negotiation, documentation, validation, and management. Yet thedetails of the process and its phases are subject tosignificant variability, and may require differentconduct depending on contextual factors related toproduct, process, project etc. Due to thesedifferences, several models of RE process have beenintroduced over the past two decades e.g. [18], [34], [35]. Likewise, numerous techniques are available toperform various RE activities, and differenttechniques are suitable for different contexts [1], [18], [19], [20], [21]. The challenge now facing REresearchers and practitioners is not the availability ofRE activities and techniques, but how to selectappropriate process activities and techniques in aspecific context.

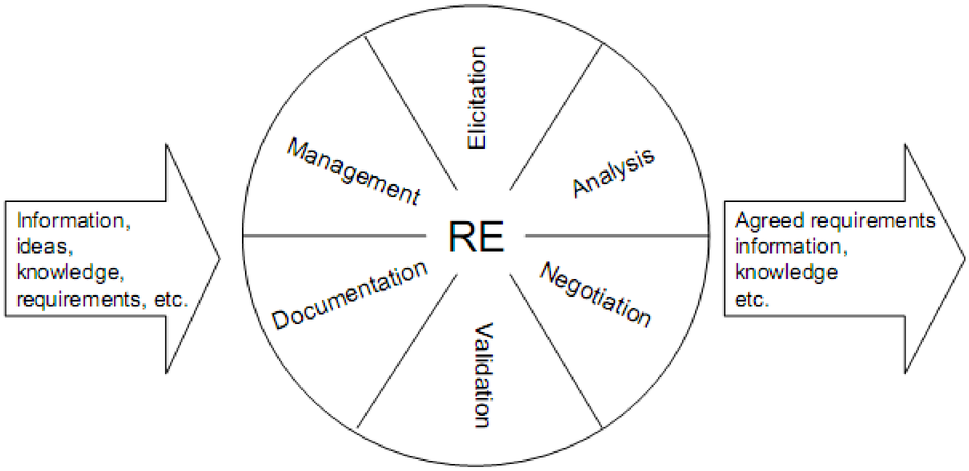


Figure 2. Generic RE Process

Various dimensions of contextdetermine the choice of specific activities andcorresponding techniques for RE by strengthening orweakening the strategic relevance of each activity and technique. For instance, for a mission-criticalproject with expert stakeholders having implicitknowledge, the elicitation activity will attain highpriority, and emphasis will be on those elicitationtechniques that facilitate communication with thestakeholders.

In practice RE activities are usually done inan ad-hoc and un-systematic manner and selection ofRE activities techniques is taken as a matter ofpersonal preferences [7], [8]. However, an ill-advisedchoice of activity/technique can lead todisappointment and waste of resources [11].

1.2. Significance of RE Decision Support (REDECS) Systems

RE is largely a decision-making process [62]. If RE decisions are systematically optimum, the probability of success in systems engineering also goes up [62]. There are significant difficulties in RE decision-making mainly because it is a knowledge-intensive activity, and that human decision-makers in general have cognitive limitations [62]. Orasanu and Connolly [63] have listed eight factors that complicate decision-making in natural settings; all of these factors are present in RE i.e. a) ill-structured problems, b) uncertain, dynamic environments, c) shifting, ill-defined, or competing goals or values, d) action and feedback loops, e) time stress, f) high stakes, g) multiple players, and h) organizational goals and norms.

In face of the above situation, there isa palpable insufficiency of automated guidance toassist the requirements engineers in systematicallydeciding which RE activities and techniques toemphasize under particular circumstances.

Suchdecision-support system should take a holistic view of activities to achieve process goals in a consistentand efficient manner. For instance, if it is decidedthat requirement change will be managed formally, appropriate preparatory steps must also be takenduring elicitation and specification.

1.3. Problem Statement

The aim of this work is to design and build an intelligent decision support system to assist the project managers, Requirements engineers in selecting appropriate techniques for various phases of Requirements Engineering process. The system should provide adaptability and expert-emulation by synergizing the relative merits of Experiential Learning and Knowledge Based Systems.

1.4. Significance of Problem

Requirements engineering is a pivotal phase of Software Development Lifecycle (SDLC). Effective RE execution saves costs and avoids time-consuming re-works. However, RE management is an art and is learnt with time and experience. Numerous techniques have been proposed for handling various dimensions of RE; however, selection of a set of most appropriate techniques in context of a specific project is a complex task due on one hand to the large number of parameters involved and on the other to the non-deterministic nature of the domain. The hypothesis of this thesis is that Artificial Intelligence techniques of Fuzzy Logic, Case Based Reasoning (CBR) and Knowledge Based Systems (KBS) can effectively handle the expert-oriented, non-deterministic, and gradually adaptive nature of decision-making in the domain.

1.5. Literature Review

Research within the field of RE decision-making is in its infancy [64]. One of the most well-known approaches is ACRE (ACquisition of REquirements) [1] by Maiden and Rugg. Considering requirements acquisition as a generic activity in Software Engineering, Knowledge Engineering, and Social Sciences, the ACRE framework could be used to evaluate acquisition techniques along six different facets. Besides providing practical guidelines, the paper also suggested incorporating ACRE in a decision support system. The question-driven guidelines of the paper are useful as checklist for RE decision making, but the paper focuses on only acquisition aspect of requirements. Beatrice [59] also presented a decision-situation modeling framework for RE.

Lobo et al [9] classified the RE process into several sub-activities, and guidelines were suggested for mapping the sub-activities to RE techniques. However, the paper did not elaborate on how to actually implement the guidelines. Similarly, Kheirkhah [31] also discussed important factors for selecting RE techniques. This work, too, is theoretical in nature.

Zhang et al [11] deal specifically with one specific phase of RE i.e. Requirements Elicitation. They categorized requirements elicitation techniques into four groups: conversational, synthetic, observational, and analytic. The authors compared strengths and weaknesses of different categories and explained applicability of each category for various situations.

Bickerton et al [24] presented a classification of RE techniques on the basis of social structure in an organization. However, no decision support techniques were discussed.

Gunda [12] presented experimental case study to bring out a comparative analysis of various elicitation techniques. The paper concluded that choice of elicitation technique should be based on application type and the situation.

Kausar et al [13] analyzed and compared various requirement elicitation techniques, and proposed a linear mathematical model evaluation of elicitation techniques.

Tsumaki et al [2] presented a framework to match RE techniques with project characteristics. A two-dimensional space was devised to categorize RE techniques with the horizontal axis describing how the RE process is conducted and the vertical axis describing the target object of process. Range of values for the horizontal axis spanned lied extremes of static and dynamic, whereas for the vertical axis the extremes were expressed by labels closed and open. Depending on the position a particular project context occupies in this space, the appropriate RE technique is suggested in the form of guidelines. The paper did not provide a practical implementation of the proposed decision-making framework.

Hickey et al [21] proposed a state based iterative model for selection of requirements elicitation techniques. The model highlights the importance of knowledge in proper execution of requirements elicitation process.

From a computational perspective, the most ground-breaking work has been done by Jiang and Eberleinet al. In [65] the authors presented cluster analysis to compare and classify RE techniques and to relate them logically to project attributes.Jiang et al [3] presented a Knowledge-based Approach for Selection of Requirements Engineering Techniques (KASRET). A library containing knowledge about different RE techniques was maintained as part of Requirements Engineering Process Knowledge Base (REPKB). Case-Based Reasoning (CBR), Frame Based Reasoning (FBR), and Relational Reasoning approaches were proposed for selecting appropriate RE technique in the light of project characteristics and experiential knowledge.

1.6. Discussion

There are very few approaches for REDECS that are based on formal analysis of various RE techniques and relating the techniques to their context of use signified by project and process parameters. Further, most of the approaches discussed above deal with only a few phases (mainly only one i.e. requirements elicitation) of RE [11], [12], [13], [21]. As a result, only limited support is provided for RE decision making. Moreover, there is limited role ascribed to knowledge (generic as well as specific) and intelligence, which is the primary contribution of this thesis.

1.7. Contributions of the Thesis

Following contributions are made by the thesis work in field of SE in general, and RE in particular:

i. A DECS system is presented to support technique selection for all phases of RE simultaneously.

ii. A synergistic AI framework for RE decision making is presented based on Fuzzy Logic, Case Based Reasoning, and Knowledge Based System.

iii. The system enhances experiential learning with expert knowledge for better decision making.

iv. KEEREDECS framework is inherently capable of handling abstract concepts of the domain using fuzzy linguistic similarities.

v. KEEREDECS framework is intended to address the knowledge-intensive and human-centered nature of decisions in RE by virtue of generic domain knowledge as well as knowledge accumulating with experience.

1.8. Organization of Thesis

This thesis is arranged in five chapters. In the next chapter, we present a survey and analysis of prominent literature related to the problem and presents the theoretical background of the workdone so far in this area. Chapter 3 describes the proposed KEEREDECS framework in detail. Chapter 4 gives experimental results of the prototype system. Chapter 5 concludes the thesis along with suggestions for future work.

CHAPTER

2. Theoretical Background

2.1. Artificial Intelligence (AI) and its Role in Software Engineering

AI is a study of computations that make it possible to reason, perceive, and act [45]. AI has a rich repertoire of concrete and systematic techniques like Case Based Reasoning (CBR),Agent Oriented Systems (AOS), Knowledge Based Systems (KBS), and Computational Intelligence (CI).

Recently there has been growing interest in research on intersection of SE and AI.For instance, KBS are being used for Learning Software Organizations (LSO) and SE Knowledge Engineering. CI is being used in knowledge discovery [50], and Software Quality Assurance [49].CBR is being applied to implement the Experience Factory (EF) concept of software organizations where an EF is a setup for experience maintenance and reuse.A prominent application of EF has been in NASA SE laboratory.

A number of books, journals, and conferences have been launched with the specific intent to develop the state of AI-SE synergy.Examples include international conference on Automated Software Engineering (ASE), workshop on Soft Computing Applied to SE (SCASE), workshop on Intelligent Technologies for SE (WITSE), the International Journal on Software Engineering and Knowledge Engineering (IJSEKE), International Journal on Knowledge and Information Systems (KAIS) etc.

There is astrong correspondence between SE and AI.SE is a practical field that depends on human judgment and expertise forsuccessful execution.Attempts to automate the process of decision making in SE have to face challenges in the form of a large number of domain parameters with uncertain relationships and ambiguous scales of measurement. AI techniques offer support in SE decision making byemulatinghuman decision makingprocesswith basic attributes of approximate reasoning, learning, and knowledge accumulation.Systematic software development including Requirements Engineering, Designing, and Project Management can be achieved through application of appropriate AI techniques to build intelligent systems for the purpose.

Below, we introduce various AI techniques that are relevant to the present work:

2.2. Case Based Reasoning (CBR)

SE is a dynamic field, where often the best practices and lessons are learnt with time and experience. Reuse of experience is an important element of SE practice [48]. CBR provides a suitable implementation framework for this experience reuse [46]. Case-based reasoning (CBR) is based upon the fundamental premise that similar problems are best solved with similar solutions [51]. The idea is to learn from experience. CBR isdifferent frommany other artificial intelligence techniques in that it is notmodel based. This means, unlike knowledge based approaches that use rules, thedeveloper does not have to explicitly define causalities and relationshipswithin the domain of interest [51]. For poorly defined problem domains this is a major benefit.

A CBR system is a four-stage process based around an experience repository called the case base. Each case comprises two parts; a description part and asolution part. The description part is normally a vector of features that describe the case state at the point at which theproblem is solved. The solution part describes the solution for the specific problem.

CBR is actually a four-stage process, called R^4 model [52] as shown in figure 3:

Retrieve: The most similar case as compared to the query case is retrieved from the case base on basis of case feature similarity.

Reuse: The retrieved case is applied to solve the query case in light of the solution part of the retrieved case.

Revise: The retrieved case is adapted to better fit the features of the query case.

Retain: The revised case is stored in the case base for future reference.

Figure [51] [52] illustrates the process.

One limitation of CBR is that it cannot deal with missing values of attributes [53]. Moreover, similarity definition for comparison of case attributes is not straightforward. While there is debate on which similarity measure to use for numeric attributes, even more formidable challenge is faced while comparing categorical attributes which are so common in SE.

In SE, CBR has been used for prediction and reuse. Prediction involves cost prediction and quality prediction [56] whereas reuse applies to design patterns, processes, and project experiences [52], [53], [54], [55].

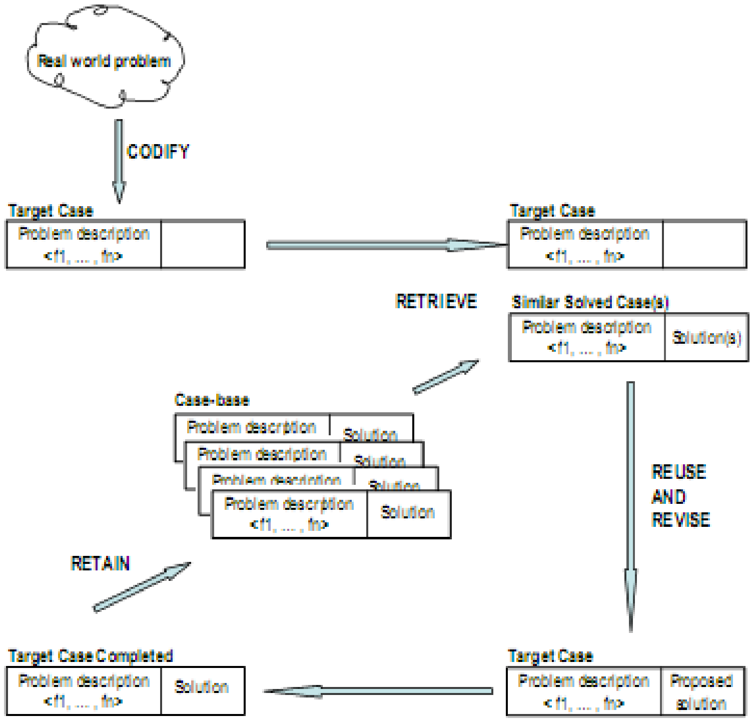


Figure 3. CBR Process [52]

2.3. Knowledge Engineering(KE)

It is beyond human ability to remember and completely comprehend all knowledge of a field or problem domain. Thus, ways have to be found so that existing knowledge can be easily accessed and used intelligently. Knowledge engineering is a discipline that can provide constructive and productive help for SE problems. Knowledge engineering is a subfield of AI that produces a type of computer system called Knowledge Based Systems (sometimes also known as Expert Systems). KBS are computer programs designed to perform tasks usually done by human experts, or to solve problems that are beyond the capability of conventional computer systems. KBS use a knowledge base of application domain or application task in a reasoning and inference framework to solve a given problem.

Although there are many varieties of KBS, the basic structure of a KBS is similar and shown in Figure 4. There are the following basic elements:

Knowledge base: The knowledge base is the heart of the KBS. It contains relevant, domain-specific problem solving knowledge such as:

- Rules. They may consist of heuristic rules or production rules. These rules are the formalized knowledge of the domain and provide a foundation for the generation of new knowledge.
- Data. These data include the facts, objects, structures and properties of the objects, and the relationships between them. The construction of the knowledge base, its usage and maintenance is one of the essential parts of the overall process of knowledge engineering.
- Inference engine.The inference engine is a program that conducts the reasoning process for solving problems based on the knowledge contained in the knowledge base. It interprets the knowledge and derives a solution of the problem. The reasoning process can generate additional knowledge from already existing rules and facts.
- User interface.User interface provide the links between the system and users. There can be many types of interfaces, such as textual, graphical, menu-driven, etc.

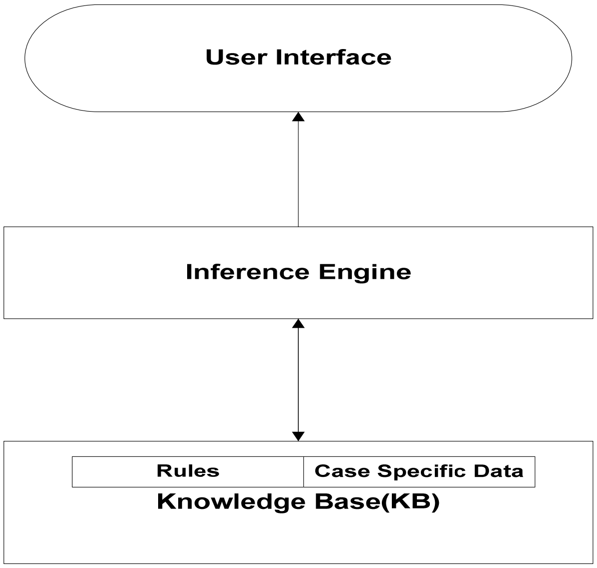


Figure 4. Basic structure of a KBS

KBS represent a complementary alternative to CBR in terms of knowledge. While CBR cases represent experiential knowledge on specific instances of a domain, KBS encapsulates expert specified rules to respond to several generic situations.Whereas CBR bypasses knowledge elicitation, KBS offers the advantage of more concise knowledge representation while facing the challenge of so-called knowledge-acquisition bottleneck. KBS have been applied to knowledge management in SE [47].

2.4. Computational Intelligence (CI)

CI is theterm applied to a number of synergistic techniques including Fuzzy Logic [23], Neural Networks[68], andEvolutionary Algorithms [69] thatemulate characteristicsof perception, learning, and decision-making in natural species. Fuzzy logic is an extension of classical Boolean logic where the nature ofset membership can be described in terms of degrees of truth instead of absolute true or false.). For example, a software product might be described as a large product to a degree 0.8, in other words it is believed to be quite large. Interestingly, fuzzy set membership may also overlap so the same component might also be described as medium to a degree of 0.6. A fuzzy set F is of the form F: U→[0,1] where U is the universe of discourse. A fuzzy set can have any shape as appropriate to the concept being modeled. For instance, Figure [70] shows the fuzzy concept “young” as applied to a person’s age.

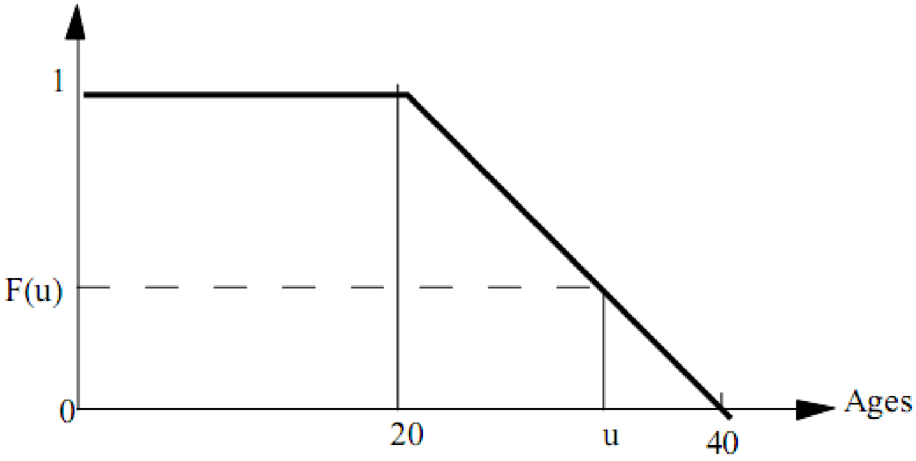


Figure 5. Fuzzy Set "Young"

The linguistic fuzzy model is built upon a set of IF-THEN rules which together with an inference engine determine response of the model to fuzzified inputs. The system output can optionally be defuzzified to obtain a crisp response. Fuzzy logic is an ideal tool for dealing with uncertainty, vagueness, and imprecision as it happens in real life dynamic situations.

2.5. Synergizing CBR, KBS and CI

Each of the AI techniques described above has its own strengths and caveats. It has been observed that systems combining two or more of these techniques can perform better than any of the individual constituents [37], [58]. While integration of CBR and KBS can consolidate knowledge representation and learning, incorporation of Fuzzy Logic can enable linguistic representation and support approximate reasoning. An example is FCBR technology which classify software components into low and high levels of defects [57]. In context of a military command and control software project, they employ fuzzy rather than crisp values to describe case features coupled with fuzzy logic to assess similarity.

2.6. Summary

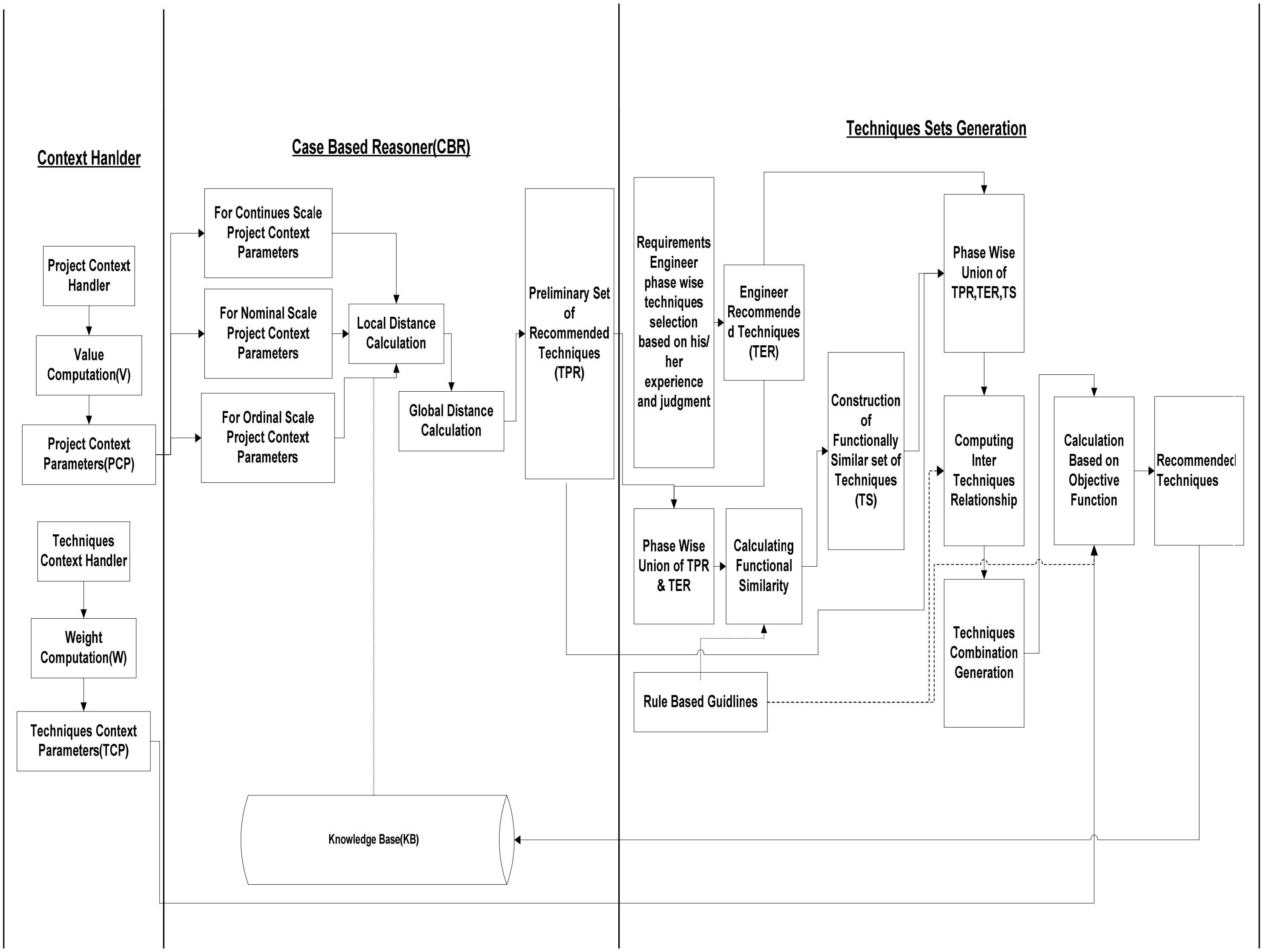
In recent years the collaboration of AI techniques like Case Based Reasoning (CBR), Agent Oriented Systems (AOS), Knowledge Based Systems (KBS), Computational Intelligence (CI), Fuzzy Logic [23], Neural Networks [68], and Evolutionary Algorithms [69] with SE has breakthrough in decision making systems. This intersection between AI and SE has largely overcome the hurdles faced by in the form of a large number of domain parameters with uncertain relationships and ambiguous scales of measurement during all phases of software development process i.e.; Requirements Engineering, Designing, Development, Testing, Deployment, Maintenance and Project Management.

CHAPTER

3. KEEREDECS Framework

3.1. Reference RE Process Model

In the proposed framework, the RE process model of Kotonya and Sommerville [18] has been adopted.Themodel has four elements: elicitation, analysis/negotiation, documentation, and validation of requirements. KEEREDECS is intended to propose most suitable techniques for each of these activities in the context of a specific project. A complete architecture of KEEREDECS is shown in figure 6.



Context Hanlder
Case Based Reasoner(CBR)
Techniques Sets Generation
Project Context Handler
Value Computation(V)
Project Context Parameters(PCP)
Techniques Context Handler
Weight Computation(W)
Techniques Context Parameters(TCP)
For Continues Scale Project Context Parameters
For Nominal Scale Project Context Parameters
For Ordinal Scale Project Context Parameters
Local Distance Calculation
Global Distance Calculation
Preliminary Set of Recommended Techniques (TPR)
Knowledge Base(KB)
Requirements Engineer phase wise techniques selection based on his/ her experience and judgment
Engineer Recommended Techniques (TER)
Phase Wise Union of TPR & TER
Calculating Functional Similarity
Rule Based Guidlines
Construction of Functionally Similar set of Techniques (TS)
Phase Wise Union of TPR,TER,TS
Computing Inter Techniques Relationship
Techniques Combination Generation
Calculation Based on Objective Function
Recommended Techniques

Figure 6. KEEREDECS Architecture

3.1.1. Context Handler

The context handler handles context parameters which are related to project, team, stakeholder, constraints, process, and product. The context parameters have been selected from historical data after extensive literature survey [2], [8], [13], [25], [29], [30], [31] as well as discussions with industrial experts and researchers. Project context attributes are divided into three major categories on basis of data type: Continuous-valued, Ordinal-valued/Absolute-Valued, and Nominal scale valued.

Continuous-valued attributes are those whose values vary within a continuous numeric range. Ordinal scale parameters are the one which values are categorical, yet there is an order in the range e.g. {Low, Medium, High}. Nominal scale attributes are purely categorical attributes with no clear-cut comparison between categories, e.g. the possible values of industry domain expressed as {web, standalone, embedded}. Details of project context parameters is given in Appendix B.

The context handler obtains project context parameters and passes them to the next module in the following form:

PCP = {V1, V2, V3,…, Vn}

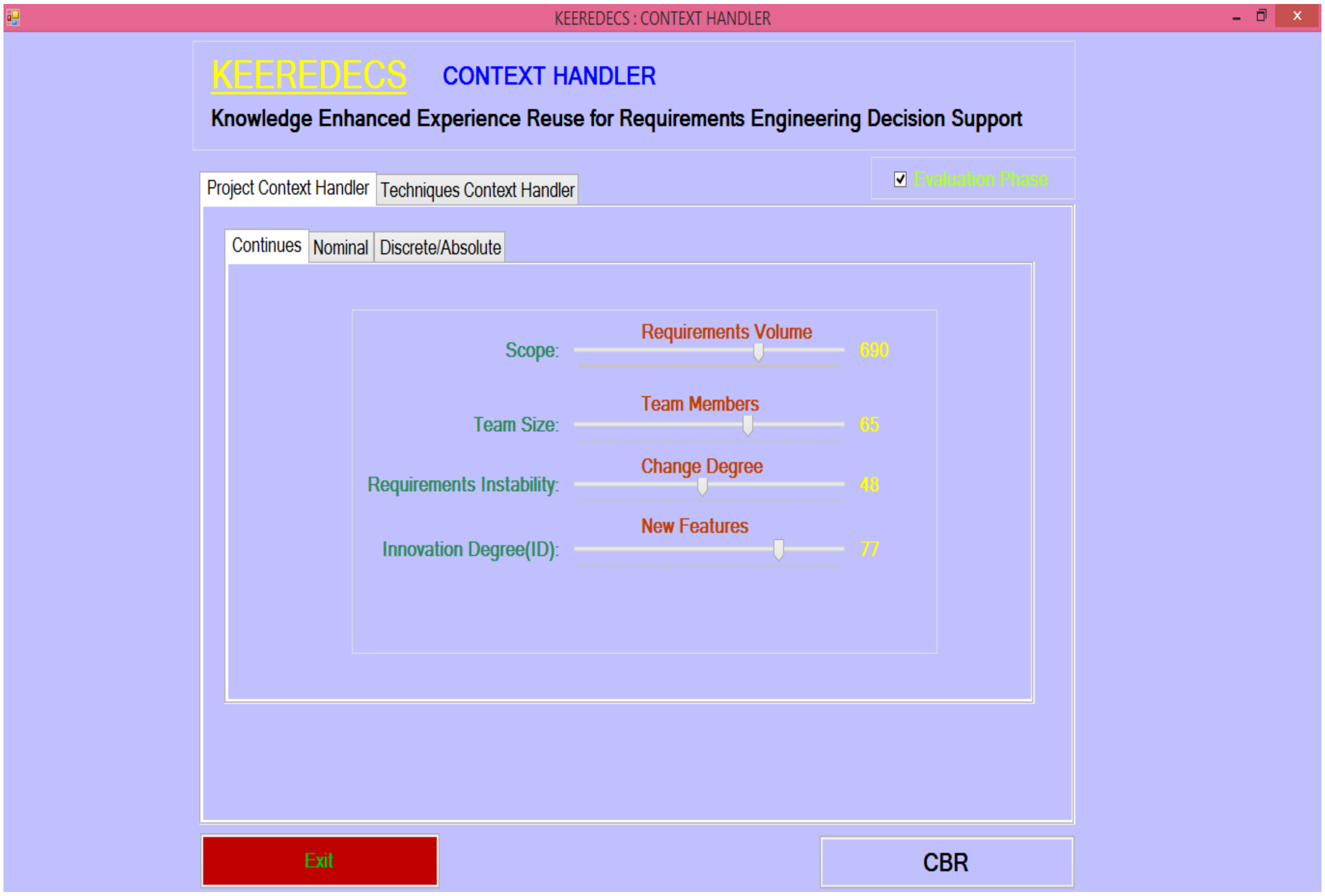
WhereV denotes (normalized) value each project context parameter and n is the total number of project context parameters.

Besides the project attributes, the context also includes importance (weight- a Boolean value) of each technique attribute as decided by the requirements engineer. The techniques attributes and support of each technique for these attributes have been chosen after extensive literature survey [2], [8], [13], [25], [29], [30], [31] as well as discussions with industrial experts and researchers. Details of project context parameters are given in Appendix C.

Mathematically we can write it down as under:

ACP = {W1, W2, W3,…,Wn}

Where n is the total number of techniques attributes and W denotes weight of each technique attributes. System development of context handler utility is shown in figure 7.


KEEREDECS : CONTEXT HANDLER
KEEREDECS
CONTEXT HANDLER
Knowledge Enhanced Experience Reuse for Requirements Engineering Decision Support
Project Context Handler
Techniques Context Handler
☑ Evaluation Phase
Continues
Nominal
Discrete/Absolute
Exit
CBR

Scope: Requirements Volume 690

Team Size: Team Members 65

Requirements Instability: Change Degree 48

Innovation Degree(ID): New Features 77

Figure 7 : KEEREDECS Context Handler

3.1.2. Fuzzy Case Based Reasoning (FCBR) Module

The main purpose of FCBRis to calculate the similarity between current case and all the cases in the case base on the basis of project context parameters values.

Similarity Calculation:

For calculating similarity between a query case and a case in the case base, first the local similarities between individual case (context) parameters are computed, and then they are integrated in a global similarity measure. Local similarity depends on data type of the attribute, and is explained below.

For project context parameters, similarity value is calculated as:

$$Sim(Xi,Y) = (\sum_{j=1}^{k} F^{m}(Xi,j\,,Yj))/k \text{ ------------- } (3.1)$$

Here $F^{m}(Xi,j\ ,Yj)$ is a membership grade of two project context parametersin configured fuzzy sets.

We have configure fuzzy sets to measure similarity between two attributes. As the expert describes fuzzy relationships between attributes using fuzzy words, corresponding fuzzy sets are constructed to emulate expert comparison. Thus,expert imparts their sense of discrimination via fuzzy words into fuzzy sets. Thus, we acquire knowledge about how to compare three kinds of attributes: Ordinal,continuous, and nominal attributes. This acquired knowledge is representedbelow in three functions—$F^{\circ}(Xi,j\ ,Yj)$, $F(Xi,j\ ,Yj)$, $F(Xi,j\ ,Yj)$.

For ordinal scaled attributes, we have used mathematically equations proposed in [43], [44] for local distance calculation between two ordinal scale project context parameters using configured fuzzy distance set shown in Table 1. It is because the distance between two ordinal scale parameters will be assigned a membership grade. The higher the grade lesser distance is between attributes and vice versa.

For calculating distance between nominal scale project context parameters we have used classic fuzzy rule base approach refers as a “fuzzy associative memory” by Kosko (1997). How they differ is in the particular fuzzy rules elicited from experts. Knowledge acquisition can be performed by having the expert fill in a questionnaire. Classic Fuzzy Rules Block for Organization Customer Relationship nominal parameter is given below while for rest of nominal parameters it is given in Appendix G.

Table 1. Distance calculation for ordinal scaled parameters

$F^{\circ}(Xi,j\ ,Yj)$	Yj					
Xi,j		**Very High/ Very Big (5)**	**High/Big (4)**	**Medium (3)**	**Low/Small (2)**	**Very Low/ Very Small (1)**
	Very High/ Very Big (5)	1.00	0.75	0.5	0.25	0.00
	High/Big (4)	0.75	1.00	0.75	0.5	0.25
	Medium (3)	0.5	0.75	1.00	0.75	0.5
	Low/Small (2)	0.25	0.5	0.75	1.00	0.75
	Very Low/ Very Small (1)	0.00	0.25	0.5	0.75	1.00

Table 2 : Classic Fuzzy Rules Block for Business Client Association

```
FunctionRules_Block_Bus_Client_Association($Xi, j$,$Yj$)
VAR_OUTPUT
  Membership_Grade : REAL;    // Membership Grade of I/P Parameters
END_VARd
RULE1 :
IF$Xi, j$ISITTAND$Yj$ISSCRTHEN
Membership_Grade IS0.75
RULE2 :
IF$Xi, j$ISITTAND$Yj$ISDGPTHEN
Membership_Grade IS0.50
RULE3 :
IF$Xi, j$ISITTAND$Yj$ISTPTHEN
Membership_Grade IS0.25
RULE4 :
IF$Xi, j$ISITTAND$Yj$ISRNSBFTHEN
Membership_Grade IS0.25
RULE5 :
IF$Xi, j$ISSCRAND sc ISITTTHEN
Membership_Grade IS0.75;
RULE6 :
IF$Xi, j$ISSCRAND$Yj$ISDGPTHEN
Membership_Grade IS0.25;
RULE7 :
IF$Xi, j$ISSCRAND$Yj$ISTPTHEN
    Membership_Grade IS 0.25;
RULE8 :
IF$Xi, j$ISSCRAND$Yj$ISRNSBFTHEN
Membership_Grade IS 0.00;
RULE9 :
IF$Xi, j$ISDGPAND$Yj$ISITTTHEN
```

Membership_Grade **IS**0.50;

RULE10 :

IF$Xi, j$**IS**DGP**AND**Yj**IS**SCR**THEN**

Membership_Grade **IS**0.25;

RULE11 :

IF$Xi, j$**IS**DGP**AND**Yj**IS**TP**THEN**

Membership_Grade **IS**0.50;

RULE12 :

IF$Xi, j$**IS**DGP**AND**Yj**IS**RNSBF**THEN**

Membership_Grade **IS**0. 25;

RULE13 :

IF$Xi, j$**IS**TP**AND**Yj**IS**ITT**THEN**

Membership_Grade **IS**0.25;

RULE14 :

IF$Xi, j$**IS**TP**AND**Yj**IS**SCR**THEN**

Membership_Grade **IS**0.25;

RULE15 :

IF$Xi, j$**IS**TP**AND**Yj**IS**DGP**THEN**

Membership_Grade **IS**0.50;

RULE16 :

IF$Xi, j$**IS**TP**AND**Yj**IS**RNSBF**THEN**

Membership_Grade **IS**0.25;

RULE17 :

IF$Xi, j$**IS**RNSBF**AND**Yj**IS**ITT**THEN**

Membership_Grade **IS**0.25;

RULE18 :

IF$Xi, j$**IS**RNSBF**AND**Yj**IS**SCR**THEN**

Membership_Grade **IS**0;

RULE19 :

IF$Xi, j$ **IS**RNSBF**AND**Yj **IS**DGP**THEN**

Membership_Grade **IS**0.25;

RULE20 :

```
IF$Xi,j$ ISRNSBFAND$Yj$ ISTPTHEN
Membership_Grade IS0.25;
End Function
```

For each continuous attribute, $(Xi,j\ ,Yj)$, the expert specifies a value of Ci which is the threshold for considering two such attributes to be near each other. A fuzzy set is constructed accordingly as shown in Figure 8. Comparing two homogeneous, continuous attributes with a fuzzy set, as shown in Figure 2 [72], is a basic application of fuzzy sets.

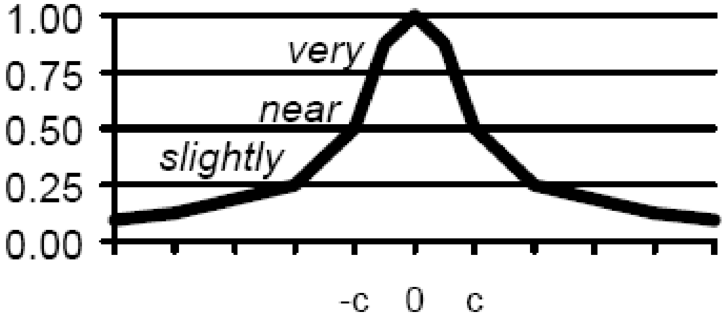


Figure 8: Fuzzy set for continuous-number attributes [72]

As a result of the above mathematical computation a single similarity (Comparative) value of each case attribute in experience base against the new case attribute will come. Suppose there are n attributes in the case, i.e. Ci = {ai1, ai2, ai3,…, ain} and the similarity of each individual attribute is calculated a simij. Then aggregate similarity between Ci and the query Cq Is calculated as per formulae (3.1). The higher similarity case(s) will be selected as best matching plan case study with the query case study. The number of cases selected using Fuzzy K-NN (Nearest Neighbors) algorithm depends on the value of Alpha (α). Complete flow of Fuzzy K-NN algorithm is shown in in figure 8.

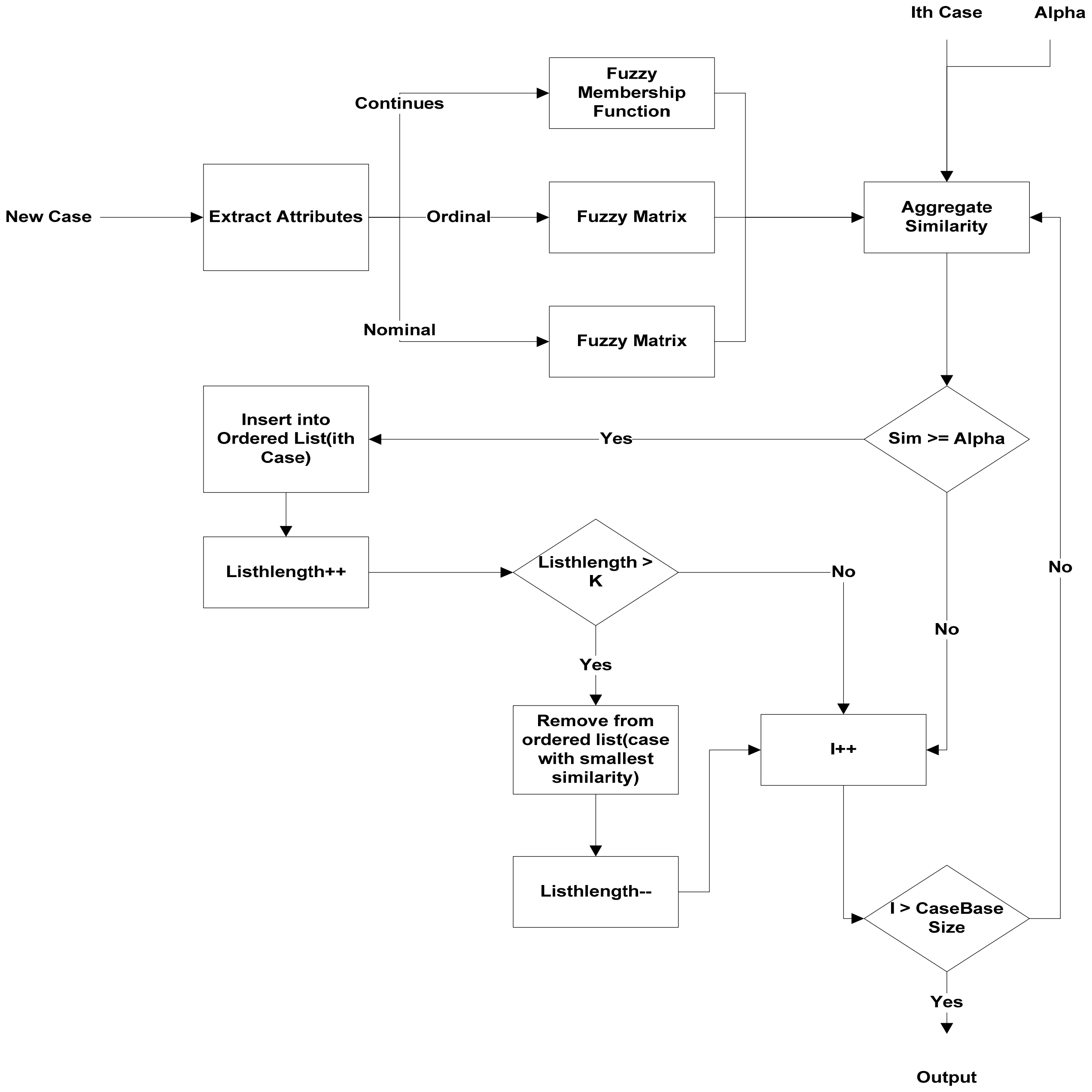


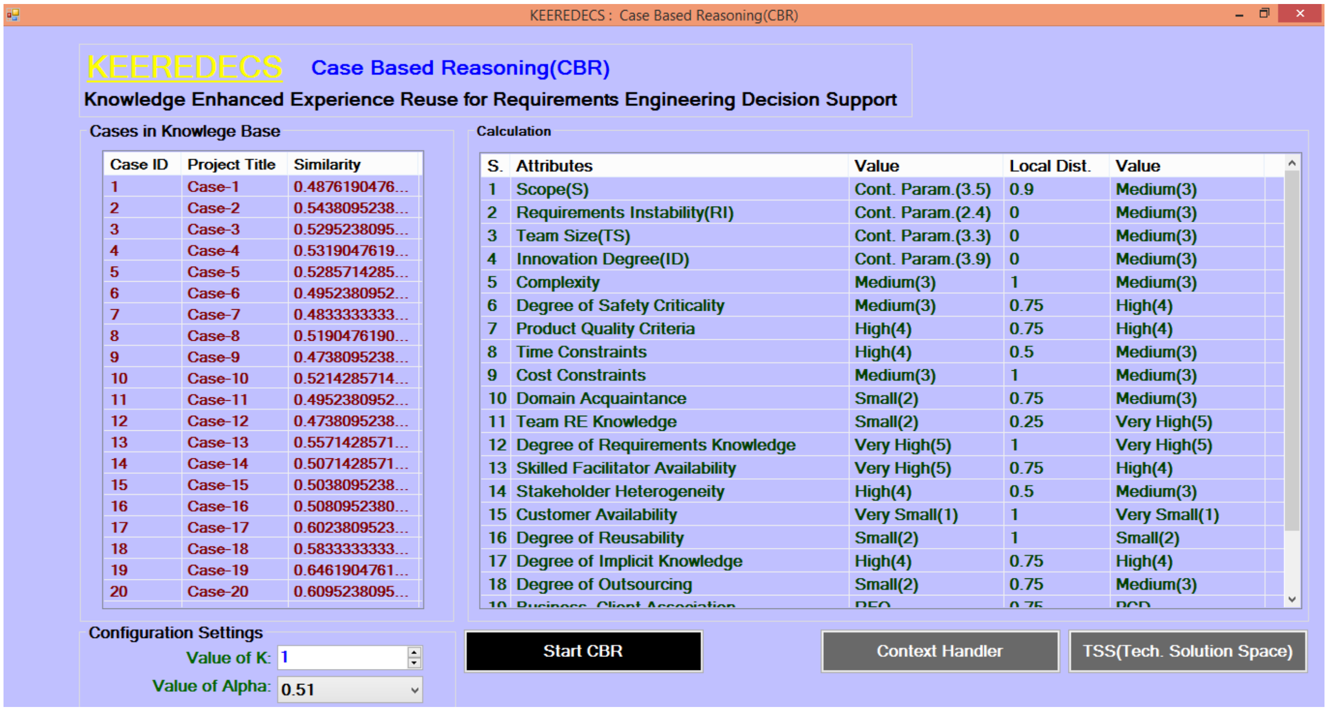
Figure 9: Fuzzy K-NN (Nearest Neighbors) Algorithm

Retrieval:

The above described similarity/distance functions are used to retrieve best matching cases from the case base with respect to the query case. In this way, asuper set **TPR** (Preliminary set of recommended techniques) will be generated, consisting of following four subsets (each corresponding to an individual phase of reference RE process).

TPR = {TPRe, TPRa, TPRd, TPRv}

where TPRe = {T1, …,Tn}, TPRa = {T1, …,Tn}, TPRd = {T1, …,Tn}, TPRv = {T1, …,Tn} are preliminary recommended technique sets for elicitation, analysis, validation, and documentation phases respectively, with the provision that any of these can also be an empty set.System development of context handler utility is shown in Figure 10.


KEEREDECS : Case Based Reasoning(CBR)

KEEREDECS Case Based Reasoning(CBR)

Knowledge Enhanced Experience Reuse for Requirements Engineering Decision Support

Cases in Knowlege Base

Case ID	Project Title	Similarity
1	Case-1	0.4876190476...
2	Case-2	0.5438095238...
3	Case-3	0.5295238095...
4	Case-4	0.5319047619...
5	Case-5	0.5285714285...
6	Case-6	0.4952380952...
7	Case-7	0.4833333333...
8	Case-8	0.5190476190...
9	Case-9	0.4738095238...
10	Case-10	0.5214285714...
11	Case-11	0.4952380952...
12	Case-12	0.4738095238...
13	Case-13	0.5571428571...
14	Case-14	0.5071428571...
15	Case-15	0.5038095238...
16	Case-16	0.5080952380...
17	Case-17	0.6023809523...
18	Case-18	0.5833333333...
19	Case-19	0.6461904761...
20	Case-20	0.6095238095...

Calculation

S.	Attributes	Value	Local Dist.	Value
1	Scope(S)	Cont. Param.(3.5)	0.9	Medium(3)
2	Requirements Instability(RI)	Cont. Param.(2.4)	0	Medium(3)
3	Team Size(TS)	Cont. Param.(3.3)	0	Medium(3)
4	Innovation Degree(ID)	Cont. Param.(3.9)	0	Medium(3)
5	Complexity	Medium(3)	1	Medium(3)
6	Degree of Safety Criticality	Medium(3)	0.75	High(4)
7	Product Quality Criteria	High(4)	0.75	High(4)
8	Time Constraints	High(4)	0.5	Medium(3)
9	Cost Constraints	Medium(3)	1	Medium(3)
10	Domain Acquaintance	Small(2)	0.75	Medium(3)
11	Team RE Knowledge	Small(2)	0.25	Very High(5)
12	Degree of Requirements Knowledge	Very High(5)	1	Very High(5)
13	Skilled Facilitator Availability	Very High(5)	0.75	High(4)
14	Stakeholder Heterogeneity	High(4)	0.5	Medium(3)
15	Customer Availability	Very Small(1)	1	Very Small(1)
16	Degree of Reusability	Small(2)	1	Small(2)
17	Degree of Implicit Knowledge	High(4)	0.75	High(4)
18	Degree of Outsourcing	Small(2)	0.75	Medium(3)
19	Business Client Association	[illegible]	0.75	[illegible]

Configuration Settings

Value of K: 1

Value of Alpha: 0.51

Figure 10 : KEEREDECS CBR

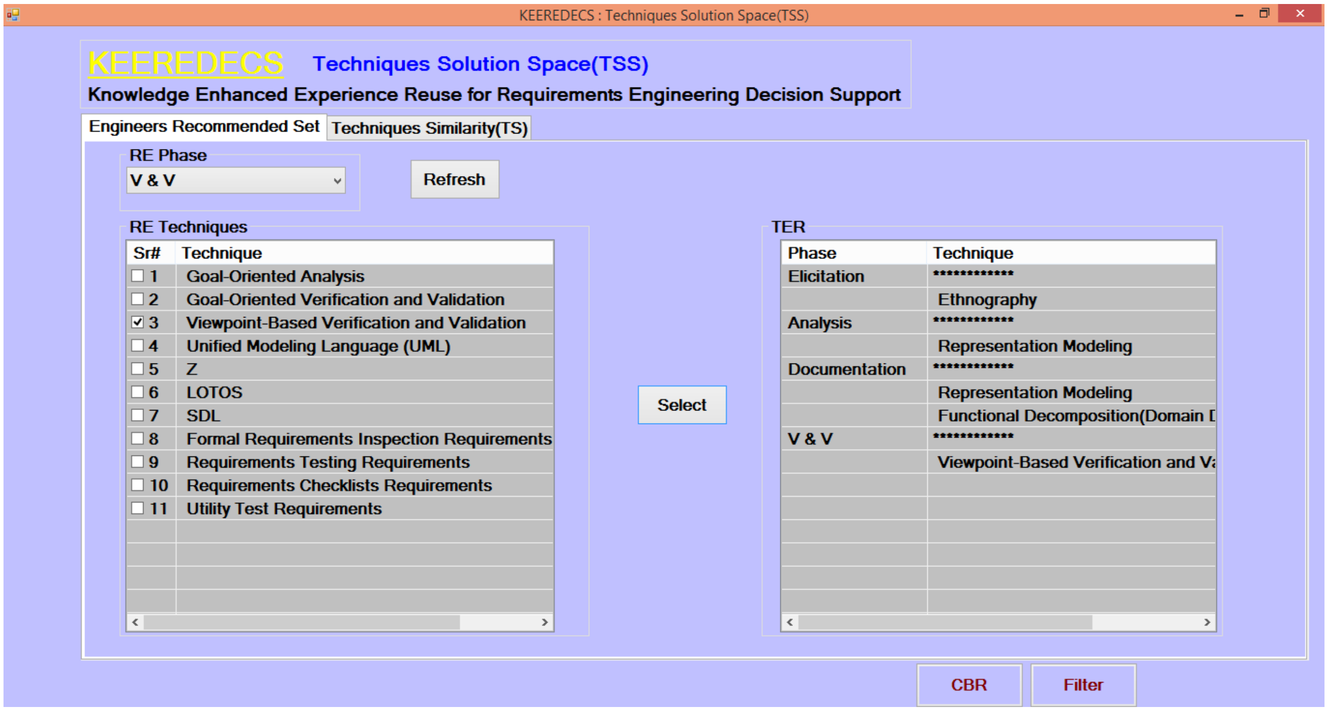
3.1.3. Expert Recommendation (Optional)

In this sub step KEEREDECS requests the Requirements Engineer to choose techniques for all of the phases of RE process based on his/her experience and judgment.

At the end of step a super set **TER** (Engineer set of recommended techniques) is generated, consisting of four subsets with each corresponding to an individual phase of RE process

TER = {TERe, TERa, TERd, TERv}

where TERe = {T1, …,Tn}, TERa = {T1, …,Tn}, TERd = {T1, …,Tn}, TERv = {T1, …,Tn}are engineer recommended techniques for elicitation, analysis, validation, and documentation phase respectively, with the provision that any of these can also be an empty set.Expert recommendation utility is shown in Figure 11.



KEEREDECS : Techniques Solution Space(TSS)

KEEREDECS Techniques Solution Space(TSS)

Knowledge Enhanced Experience Reuse for Requirements Engineering Decision Support

RE Phase: V & V

RE Techniques

Sr#	Technique
☐ 1	Goal-Oriented Analysis
☐ 2	Goal-Oriented Verification and Validation
☑ 3	Viewpoint-Based Verification and Validation
☐ 4	Unified Modeling Language (UML)
☐ 5	Z
☐ 6	LOTOS
☐ 7	SDL
☐ 8	Formal Requirements Inspection Requirements
☐ 9	Requirements Testing Requirements
☐ 10	Requirements Checklists Requirements
☐ 11	Utility Test Requirements

TER

Phase	Technique
Elicitation	************
	Ethnography
Analysis	************
	Representation Modeling
Documentation	************
	Representation Modeling
	Functional Decomposition(Domain I
V & V	************
	Viewpoint-Based Verification and Va

Figure 11 : KEEREDECS Expert Recommendation Utility

3.1.4. Techniques Sets Generation (TSG)

The repository of knowledge (KB) about RE techniques, including individual strengths, weaknesses, and inter-technique relationships is pivotal to KEEREDECS. The KB contains expert acquired knowledge about various RE techniques. Jiang et al [66] studied 46 RE techniques and detailed their attributes as making them suited for particular RE activities. The techniqueshave been selected because of their widespread use and availability of documentation. We have used the same work as foundation for creating our KB. In [65] Jiang et al presented RE technique clustering so that it is possible to group techniques as equivalent, complementary, or opposite to each other [67]. The KB in KEEREDECS enhances the same knowledge with expert knowledge for selection of RE techniques for a project. A difference of KEEREDECS from Jiang et al is that while Jiang et al's KB is completely autonomous. We have the following entities in KEEREDECS KB.

- Tbl_RET(RE Techniques)
- Tbl_REP (RE Phases)
- Tbl_RET_REP(It is a gerund b/w RE Tech. & RE Phases)
- Tbl_Attrib_Project_Levels(Project Level Context Parameters)
- Tbl_Project_Cases(Different project cases)

- Tbl_ProjCases_Attrib_Project_Levels (Project level context parameters value of different cases. It is gerund between project cases and project level context parameters)

- Tbl_Attrib_Process_Levels(Techniques Level Context Parameters)

- Tbl_RET_Attrib_Process_Levels(Support of RE techniques for different techniques level context parameters)

- Tbl_Project_Cases_Recomended_Tech (Phase wise recommended techniques for different project cases)

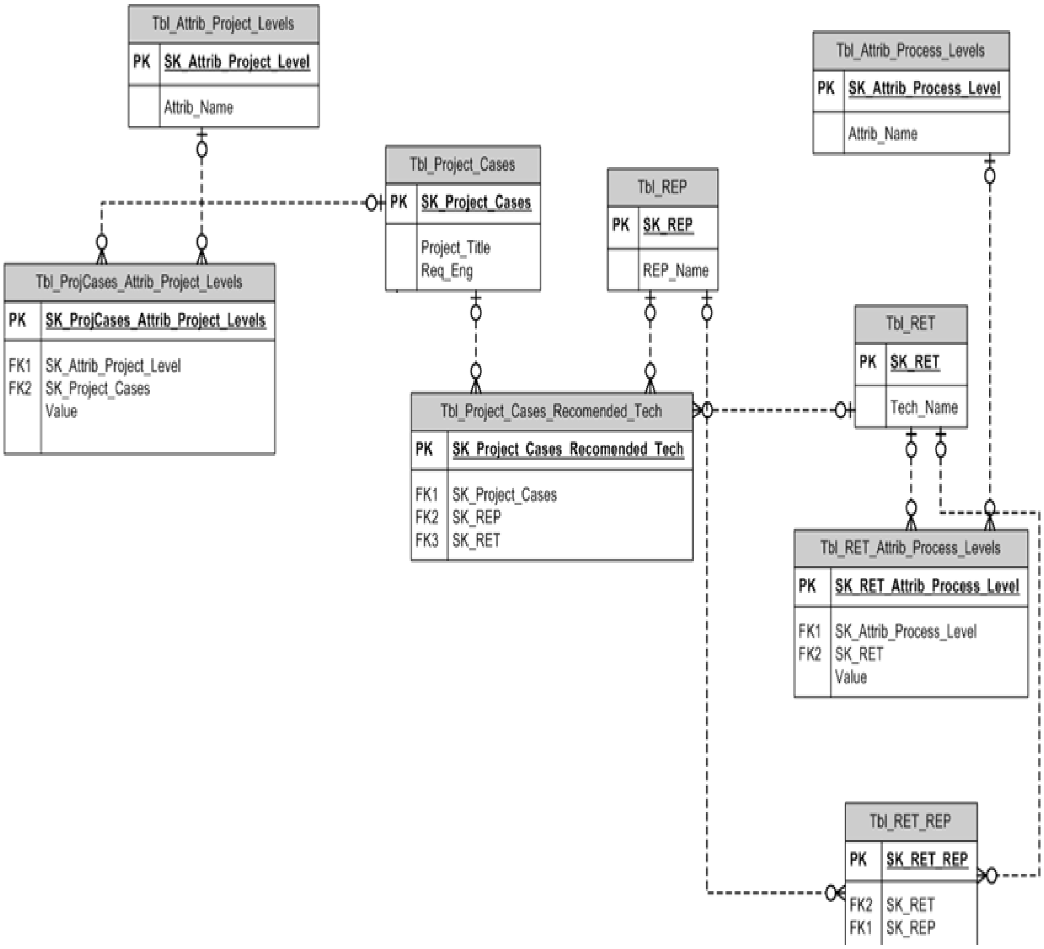

Tbl_Attrib_Project_Levels
PK SK_Attrib_Project_Level
Attrib_Name
Tbl_Attrib_Process_Levels
PK SK_Attrib_Process_Level
Attrib_Name
Tbl_Project_Cases
PK SK_Project_Cases
Project_Title
Req_Eng
Tbl_REP
PK SK_REP
REP_Name
Tbl_ProjCases_Attrib_Project_Levels
PK SK_ProjCases_Attrib_Project_Levels
FK1 SK_Attrib_Project_Level
FK2 SK_Project_Cases
Value
Tbl_RET
PK SK_RET
Tech_Name
Tbl_Project_Cases_Recomended_Tech
PK SK_Project_Cases_Recomended_Tech
FK1 SK_Project_Cases
FK2 SK_REP
FK3 SK_RET
Tbl_RET_Attrib_Process_Levels
PK SK_RET_Attrib_Process_Level
FK1 SK_Attrib_Process_Level
FK2 SK_RET
Value
Tbl_RET_REP
PK SK_RET_REP
FK2 SK_RET
FK1 SK_REP

Figure 12: KEEREDECS KB ERD

KEEREDECS allows the expert to intervene for judgment-based selection of techniques.KB developed and designed for KEEREDECS are shown in figure 9.

The KBS has three parts: data, rules, and reasoning. These are described below:

Data:

The data part is meant to contain knowledge about RE techniques. For each RE technique, the data part maintains the following information:

- Applicable RE activity
- Attribute values

Rules:

The rule part contains two types of rules; rules that describe inter-technique relationships and advisory rules. The first type of rules describes for various pairs of RE techniques how the two techniques are related to each other. The relationship may describe the techniques as similar or complementary. The relationship is generically described as a function of attribute scoring of the techniques being compared.

The advisory rules describe various scenarios under which one or another technique may be more desirable. The rules also encompass scenarios where it would be counter-productive to use a specific technique. Description of scenarios is based on project context parameters and RE technique attribute weights.

Reasoning:

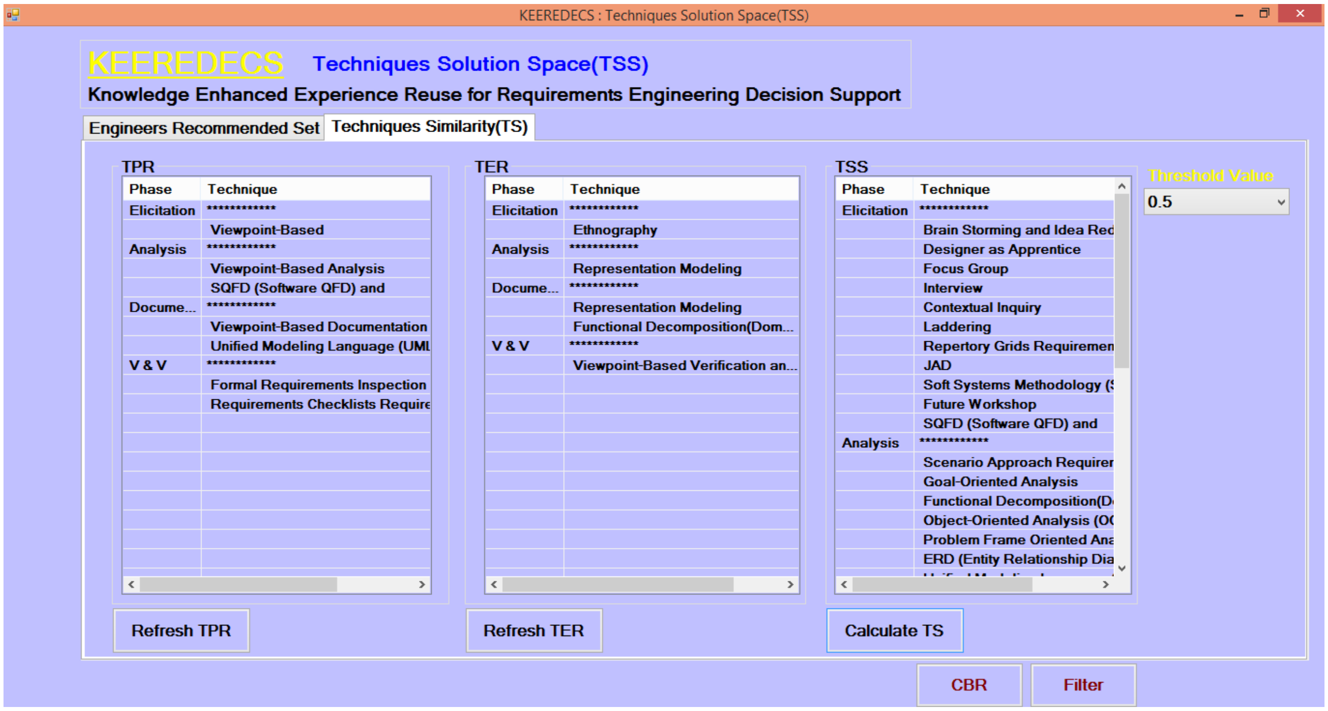
Input to this module is the union of sets TPR and TER as described in above sections.This step is meant to prune the techniques obtained by eliminating the unnecessary and inconsistent techniques.

If two or more techniques possess similar attributes, then it is sufficient and necessary to use only one of them for optimality. Two techniques T and T'are similar if and only if the sum of differences between phase wise attribute valuesof the two techniques is not more than a threshold value α. Phase wise mathematical formula are given in Appendix E.

Mathematically, we can write it as follows:

$$\boldsymbol{TS_i} = \frac{\sum_{j=1}^{n} \boldsymbol{T(Aj) - T\prime(Aj)}}{\boldsymbol{K}} <= \boldsymbol{\alpha} \text{-------------} (3.2)$$

Where$\boldsymbol{TS_i}$stands for Technique Similarity in phase i, $\boldsymbol{Aj}$ is jth attribute of the technique, K is the total number of attributes and $\boldsymbol{\alpha}$ is the threshold value provided by the user. Techniques Similarity (TS) system development is shown in Figure 13.



KEEREDECS : Techniques Solution Space(TSS)

KEEREDECS Techniques Solution Space(TSS)

Knowledge Enhanced Experience Reuse for Requirements Engineering Decision Support

Engineers Recommended Set | Techniques Similarity(TS)

TPR

Phase	Technique
Elicitation	************
	Viewpoint-Based
Analysis	************
	Viewpoint-Based Analysis
	SQFD (Software QFD) and
Docume...	************
	Viewpoint-Based Documentation
	Unified Modeling Language (UML
V & V	************
	Formal Requirements Inspection
	Requirements Checklists Require

TER

Phase	Technique
Elicitation	************
	Ethnography
Analysis	************
	Representation Modeling
Docume...	************
	Representation Modeling
	Functional Decomposition(Dom...
V & V	************
	Viewpoint-Based Verification an...

TSS

Phase	Technique
Elicitation	************
	Brain Storming and Idea Red
	Designer as Apprentice
	Focus Group
	Interview
	Contextual Inquiry
	Laddering
	Repertory Grids Requiremen
	JAD
	Soft Systems Methodology (S
	Future Workshop
	SQFD (Software QFD) and
Analysis	************
	Scenario Approach Requirer
	Goal-Oriented Analysis
	Functional Decomposition(D
	Object-Oriented Analysis (OO
	Problem Frame Oriented Ana
	ERD (Entity Relationship Dia

Threshold Value: 0.5

Figure 13 : KEEREDECS Techniques Similarity (TS) Module

Now we have three sets of recommended techniques i.e.; TPR, TER, and TS. So, the techniques suggestion space TSS can be written as:

TSS = {TPR, TER, TS}

The above threecan be best viewed in tabular/cross tab form as given below:

Table 3. Generated Technique Sets

Sets/Phases	Elicitation	Analysis	Documentation	V & V
TPR	TPRe	TPRa	TPRd	TPRv
TER	TERe	TERa	TERd	TERv
TS	TSe	TSa	TSd	TSv

During the next reasoning step vertical union of the sets will be taken for each phase. As a result of this vertical union a new set will come into existence for each phase.

Mathematically it can be shown as:

TPi= TPRe U TERe U TSe

Where TPi denotes ith phase of RE process.

After union all the techniques will be arrange in a tabular form in order to drive the relationship amongst each other.

Table 4. Inter-Technique Relationships

	T1	T2	…	Tn
T1	X	S	…	S
T2	S	X	…	O
…	…	…	…	…
Tn	S	O	…	X

Where:

X = nothing

S = Similar

O = Opposite

We can find opposite relationships between techniques using following mathematical formula:

$$\boldsymbol{TDi} = < \frac{\sum_{j=1}^{n} \boldsymbol{T(Aj) - T'(Aj)}}{\boldsymbol{K}} <= \beta\text{--------------} \ (3.3)$$

Where $\boldsymbol{TD_i}$ stands for Technique Similarity in phase **i**, $\boldsymbol{Aj}$ is jth attribute of the technique, $\boldsymbol{\beta}$ = *1* ,□ = *0.5*and **K** is the total number of attributes.Phase wise mathematical formula are given in Appendix F.

At the end different combinations of techniques are formed. The number of combinations will depend on the functionally equivalence relation (similarity) between the techniques.Mathematically,

C1 = {T1, T2, T3, … ,Tn}

C2 = {T1, T2, T3, … ,Tn}

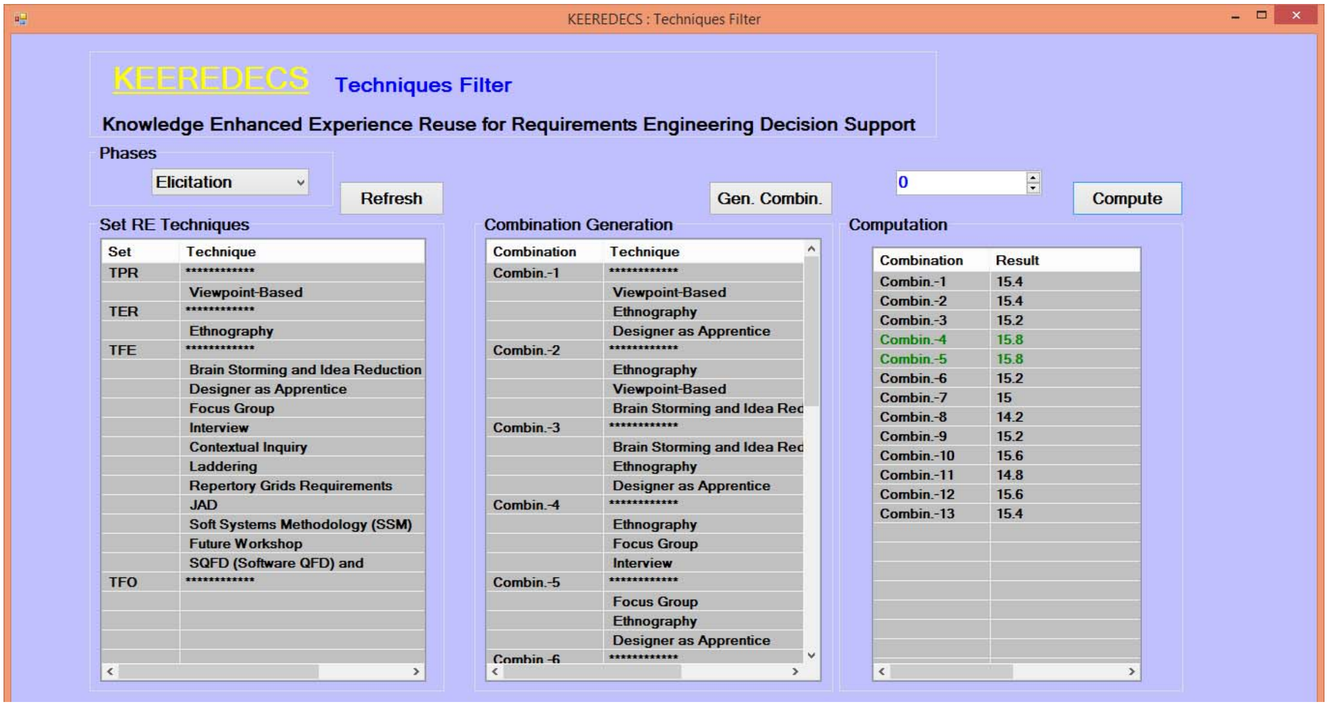
…

Cn = {T1, T2, T3, … ,Tn}

After successfully making techniques combinations each combination is passed on toan objective function to calculate its worth. The combination with highest value is selected for the correspondingphase. Phase wise objectives are given in Appendix G. The objective is given as:

$$Oi = \sum_{t=1}^{n} \left(\sum_{j=1}^{m} T(Aj) * Bj\right) \text{-------------- (3.4)}$$

Where $\boldsymbol{Oi}$ stands for Objective function for phase **i**, **n** is the total number of techniques in combination, **m** is the total number of attributes, **Aj** is jth attribute of the technique and **Bj** is the weight of an attribute.Objective Function (OF) system development is shown in Figure 14.


KEEREDECS : Techniques Filter
KEEREDECS
Techniques Filter
Knowledge Enhanced Experience Reuse for Requirements Engineering Decision Support
Refresh
Gen. Combin.
Compute

Phases: Elicitation

0

Set RE Techniques

Set	Technique
TPR	************
	Viewpoint-Based
TER	************
	Ethnography
TFE	************
	Brain Storming and Idea Reduction
	Designer as Apprentice
	Focus Group
	Interview
	Contextual Inquiry
	Laddering
	Repertory Grids Requirements
	JAD
	Soft Systems Methodology (SSM)
	Future Workshop
	SQFD (Software QFD) and
TFO	************

Combination Generation

Combination	Technique
Combin.-1	************
	Viewpoint-Based
	Ethnography
	Designer as Apprentice
Combin.-2	************
	Ethnography
	Viewpoint-Based
	Brain Storming and Idea Red
Combin.-3	************
	Brain Storming and Idea Red
	Ethnography
	Designer as Apprentice
Combin.-4	************
	Ethnography
	Focus Group
	Interview
Combin.-5	************
	Focus Group
	Ethnography
	Designer as Apprentice
Combin.-6	************

Computation

Combination	Result
Combin.-1	15.4
Combin.-2	15.4
Combin.-3	15.2
Combin.-4	15.8
Combin.-5	15.8
Combin.-6	15.2
Combin.-7	15
Combin.-8	14.2
Combin.-9	15.2
Combin.-10	15.6
Combin.-11	14.8
Combin.-12	15.6
Combin.-13	15.4

Figure 14 : KEEREDECS Objective Function Utility

1.1. Summary

It is evident from KEEREDECS architecturethat we have includedall the aspects from context project context to process values computation and from fuzzy set theory, crisp set theory to mathematical calculations to get the best decision supporting results for each phase of software requirements engineering process. We take on project and RE techniques context in first step. In next step we use Fuzzy K-NN algorithm to get the best similar case(s) as per value of K. Afterwardsstep helps us to find out functional equivalency of techniques in KB based on union of TPR and TER. In last step we makes different unique sets of techniques and pass those sets one by one through a filter to get the highest similarity.

CHAPTER

2. Analysis and Validation

Two different methods have been adopted for validating the proposed framework of KEEREDECS. First, we did a case study validation of the system using an actual industrial scenario concerning Software Engineering. Second, we validated the system using a survey of experienced Requirements Engineers. Both the methods and their findings are detailed below:

2.1. Validation using Case Studies Approach

The proposed KEEREDECS approach was validated through comparative study of two projects from an actual organization. The organization is a large sized government organization whose name has been kept anonymous. Two Projects have been studied which we will call Project-I and Project-II are almost of the same nature with few differences in context parameters. For Project-I the RE techniques recommendation was done in conventional way based on experience and expert judgment while in case of Project-II RE techniques recommendation was done by proposed KEEREDECS framework.

We have setup different parameters for comparison between two projects. The comparative study is shown in Table 5; based on this quantitative data it can easily be seen that in case of Project-II RE techniques recommended by KEEREDECS has much more positive impact on the later phases of the project and on the overall project compared to conventionally recommended techniques in case of Project-I.

2.2. Validation using Survey Approach

The proposed KEEREDECS approach was also validated by conducting an industrial survey (Survey Form Attached at Appendix-A). Survey is designed on the basis of Context Parameters used in the proposed model. Around 19 questions were asked by the participants. Questions were arranged in categories. Participants were asked to answer on the basis of their experience.

Twelve (16) participants from different organizations (public and private sectors) participated in the survey. The answers provided by the experts are then combined in Table 6.

Table 5. Case Study Comparison of KEEREDECS with Conventional RE Approach

Parameters		**Projects/Case Studies**	
		PROJECT-II	**PROJECT-I**
		Phase wise usage of RE Techniques	
		TEP: Focus Group, Interview, Ethnography **TAP:**Viewpoint-Based Analysis, Scenario Approach **TDP:** UML,ERD **TVP:**Formal Requirements Inspection	**TEP:**Informal Focus Group **TAP:** OO Modeling, State Machine **TDP:**Informal Documentation **TVP:**Informal Review(mostly peer to peer review)
Total number of precise requirements in the final version of requirements in the SRS		632	1050
Total Number of requirements engineer, system architect and analysts participated in the project		5	5
Total Numbers of software coders engaged		5	5
Genuine requirements insight before the start of the project		300	702
Total number of precise requirements elicited by the use of RE techniques.	Exact volume	252	357
	% of the total volume of requirements	40.01%	34.1%
Addition of total number of precise requirements during V & V phase	Exact volume	22	40
	% of the total volume of requirements	3.5%	3.8%
Modification of total number of precise requirements during V & V phase	Exact volume	97	89
	% of the total volume of requirements	15.3%	8.5%
Addition of total number of precise requirements during Design phase	Exact volume	29	81
	% of the total volume of requirements	4.6%	7.7%
Addition of total number of precise requirements during QC/Testing phase	Exact volume	18	53
	% of the total volume of requirements	2.8%	5%
Modification of total number of precise requirements after beginning of design phase	Exact volume	40	123
	% of the total volume of requirements	6.4%	11.7%
% of total requirements Modified after beginning of design phase		6.4%	11.7%
Intended project time frame		6 Months	9 Months
Real project completion time frame		Less than 8 Months	More than a year
Effort Estimation (Human Resource-Months)	Intended	60	90
	Real	70	120
Budget overrun in Effort Estimation (Human Resource-Months)	Amount	10	30
	% increase over Intended	16.6%	33.3%

Table 6. Survey Results

Category	Participants	Questions	Strongly Agree	Agree	Medium	Disagree	Strongly Disagree
Administration	**20**						
		The RE techniques recommended are appropriate for PROJECT-II project	8	12	0	0	0
		The recommended techniques was up to the mark for reduction of re-work during all phases of software development life cycle	4	16	0	0	0
		The recommended techniques was good enough to complete the project within intended time span.	8	12	0	0	0
		The final version of SRS was of high satisfaction for the both the customer and Administration	8	12	0	0	0
		The overall organization of Requirements Engineering process for PROJECT-II was much better than PROJECT-I and later projects	8	12	0	0	0
Coders	**20**						
		Both Functional and Non Functional requirements was easily mapped to coding	4	4	4	4	0
		The overall rate of change of requirements during coding phase was	12	8	0	0	0

		much less than PROJECT-I and other projects done before					
		The SRS document was precise and understandable than before	16	4	0	0	0
		RE Techniques used for requirements specification in SRS was easy to understand and was within acceptable range	16	4	0	0	0
		It was easy to trace back the requirements	8	8	0	4	0
		The recommended techniques was good enough to complete the project within intended time span	12	8	0	0	0
		The overall quality of the documentation is high	8	12	0	0	0
Requirements Engineers , Analyst and Architect	**12**						
		The main issue of the PROJECT-II project was easily addressable by the recommended techniques	6	4	2	0	0
		Requirements Modeling in SRS was easily mapped to designing/modeling constructs	4	8	0	0	0
		Usage of Modeling	2	4	6	0	0

	notation language such as unified modeling language was helpful and time saving in mapping requirements to modeling constructs					
	Inclusion of more formal RE techniques in recommended techniques would increase the quality of SRS and modeling	0	6	6	0	0
	With appropriate training struggle and cost to newly recommended techniques can be easily overcome which were never used in previous project by keeping cost and difficulty factor	2	6	4	0	0
	Requirements Engineering process was much more structured and planned than before	6	6	0	0	0
	Recommendation done by KEEREDECS framework was good enough for project type like PROJECT-II	4	6	2	0	0

It was revealed while conducting survey that some questions asked in survey might not be interpreted as the author intended; otherwise results could have been more accurate. It was likely because the author was not present with participants to attain the purposeful results.

The complete analysis of combined and aggregated data from Table 6is shown in the Table 7.

Table 7. Aggregated Survey Results

Category	Answer				
	Strongly Agree	Agree	Medium	Disagree	Strongly Disagree
Administration	**Percentage (%)**				
	36 %	56 %	0 %	0 %	0 %
Category -II: Coders	**Percentage (%)**				
	28.57 %	47.62 %	23.80 %	0 %	0 %
Category -III: Requirements Engineers, Analyst and Architect	**Percentage (%)**				
	54.28 %	37.14 %	2.85 %	5.71 %	0 %

The graphical representation of data analysis results in Table 7 can be view in Figure 10.

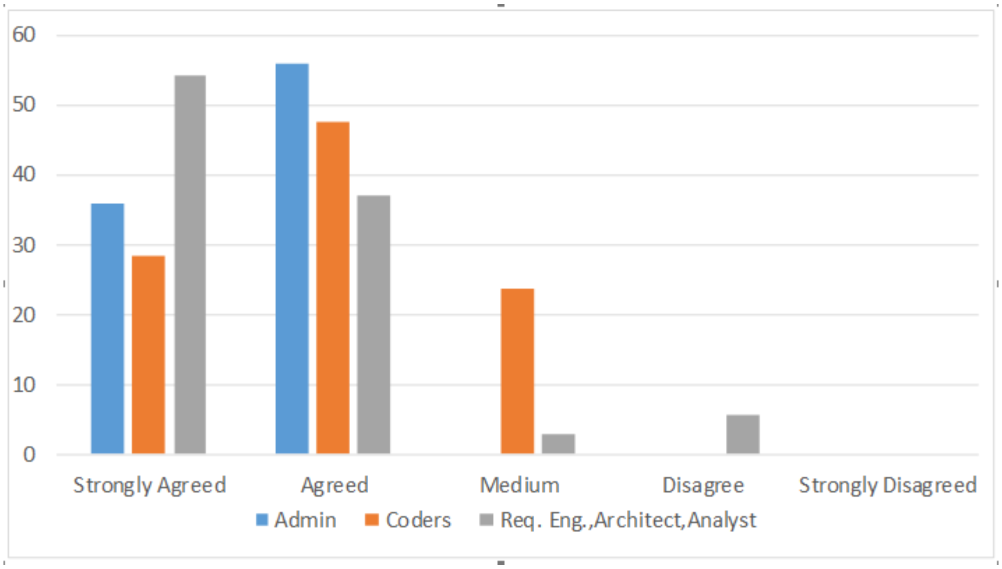


Figure 15: Survey Based Data Analysis Chart

From Figure 10 it is evident that Survey results ofKEEREDECS has average accuracy between 80% - 90%. Most of the individuals participated in the survey have shown their maximum positive interest in the "Strongly Agreed" and "Agreed". On the basis of these results we may say that the proposed framework is able to recommend RE techniques for a project with 80% - 90% accuracy. Better results can be achieved by repeatedly using the framework because it is based on knowledge engineering and experiential FCBR.

2.3. Validation Based on Value of Nearest Neighbors (K)

During last validation of our framework we have taken a phase wise techniques of a project which was successfully executed by a private firm. We have also made a thorough discussion with the industry specialist in the field of requirements engineering for those recommended techniques.

Table 8 : Validation results based on Value of Nearest Neighbors (K)

Phases/Cases	Elicitation	Techniques	Analysis	Techniques	Documentation	Techniques	V & V	Techniques	Total
TOA	36.4	1,4,5,6,15,19	27.2	12,14,28,31,38,42	41.6	14,18,20,22,23,31,38	11.2	30,37,43,44,45,46	116.4
K = 1	15.8	4,5,6	12.8	12,14,31	18.4	14,18,20,22	6.0	30,37,44	53
Difference	20.6	----	14.4	----	23.2	----	5.2	----	63.4
K = 5	16.2	1,4,5	13.4	12,14,28	19	14,18,20,31	6.0	30,37,45	54.6
Difference	20.2	----	13.8	----	22.6	----	5.2	----	61.8
K = 10	16.6	5,6,19	13.8	28,31,38	19.4	14,18,20,23	7.6	30,44, 45,46	57.4
Difference	19.8	----	13.4	----	22.2	----	3.6	----	59
K = 15	21.4	4,5,6,19	18.2	12,28,31,38	23.4	14,18,20,22, 23	7.8	30, 37,45,46	69.8
Difference	15	----	9	----	18.2	----	3.4	----	45.6
K = 20	21.8	1,5,6,19	18.4	28,31,38,42	23	14,18,20,22, 31	9.4	30,37, 43, 45,46	72.6
Difference	14.6	----	8.8	----	18.6	----	1.8	----	43.8

Main focus of this type of validation is the impact of Fuzzy K-NN algorithm on our framework. It has been revealed and it is clearly visible from Table 8 that as we increases the number of K i.e. number of nearest neighbors cases in the knowledge base, phase wise and overall difference between the standard case and the new case decreases. This proof of positive impact are also evident from Figure 11 and Figure 12.

TAO = Take Away One

Elicitation Phase:

1	Brain Storming and Idea Reduction
4	Ethnography
5	Focus Group
19	Future Workshop
6	Interview
15	JAD

Analysis Phase:

31	ERD (Entity Relationship Diagram)
28	Object-Oriented Analysis (OOA)
14	Scenario Approach Requirements
38	Unified Modeling Language (UML)
12	Viewpoint-Based Analysis
42	XP (Extreme Programming)

Documentation Phase:

22	Decision Tables
31	ERD (Entity Relationship Diagram)
20	Representation Modeling
14	Scenario Approach Requirements
23	State Machine
38	Unified Modeling Language (UML)
18	Viewpoint-Based Documentation

V & V Phase:

43	Formal Requirements Inspection Requirements
30	Goal-Oriented Verification and Validation
45	Requirements Checklists Requirements
44	Requirements Testing Requirements
46	Utility Test Requirements
37	Viewpoint-Based Verification and Validation

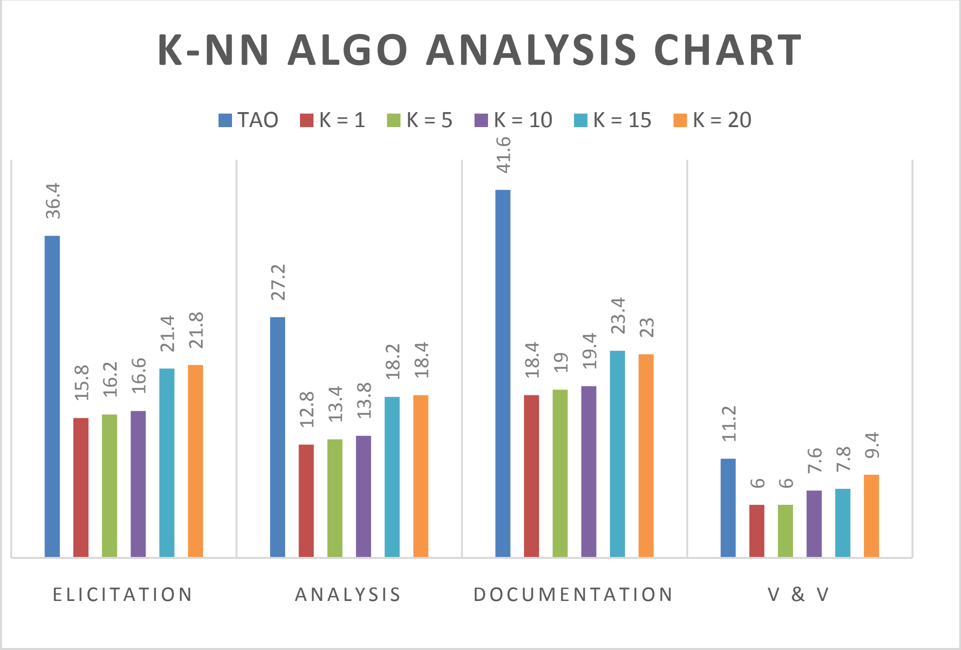


Figure 16: K-NN Algorithm Analysis Chart

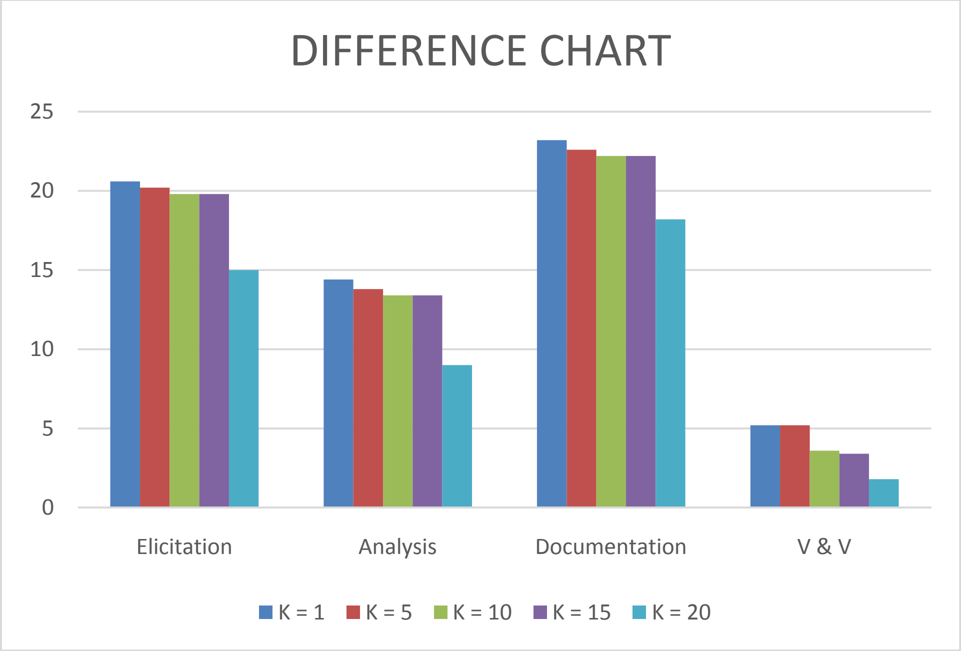


Figure 17: K-NN Difference Chart

2.4. Summary

KEEREDECS framework has been analyzed and validated using different methods and techniques. First comparative of two case studies has been done. For first case study RE phase wise techniques are recommended without using the framework while for second case study techniques were supported by the framework. Overall impact of the RE phase of the second case study were much more positive the first case study on afterwards phases of the case study.

Second approach was based on survey. A survey form was designed and filled by different professionals of the industry. Data were consolidated for analysis, it was found that most of the participants were falling their views in strongly agree and agree zone. Very few have put their views in the medium and almost none in the disagree & strongly disagree zone for KEEREDECS framework.

CHAPTER

3. Conclusions& Future Work

Requirement Engineering is a crucial process in Software Engineering and requires systematic decision making in face of many uncertainties and imprecisions. KEEREDECS presented a context based decision making framework that employs fuzzy case based reasoning to choose the most appropriate requirements engineering techniques for different phases of RE process in a given context. Requirement Engineering is qualitative domain where decision-making is judgmental and approximate. Solutions to decision-problems are nondeterministic, being dependent on complex parameters and expertise of the decision-maker. RE is rich in activities and techniques; however, notall of them are equally suitable for a particularcontext. For example, interview is suitable forrequirements elicitation but not very helpful forrequirement management and documentationactivities. Each activity and technique has associatedcosts, making it imperative for the decision-makers toselect the optimal combination of activities andtechniques.

Most of the RE decision parameters involve cognitiveuncertainty in their interpretation. While someparameters like application domain are inherentlyqualitative, others are also better handled in aqualitative manner rather than quantitative. Forinstance, complexity and scale have no standarddefinitions, and different types of metrics mayfurnish different values for these parameters.Moreover, measurement data pertaining toparameters is often incomplete, approximate, anderror-prone.The parameters in RE decision-making are related toeach other in complex ways. There is no unique wayin which the effect of various parameters and theircombinations on process performance can beexpressed. Thus crisp mathematical modeling of REtechnique selection problem does not appear to be anattractive approach.Parameter values assume different meanings underdifferent contexts, and call for different decisions. Inother words, in-exactness of the parameters istransmitted over to in-exactness of decisions based onthem. For instance, the parameter project complexitywill have different meaning for an in-experiencedteam as compared to an experienced one- and willneed different types of decisions.RE is an art that is learned with experience.Requirements Engineers develop the insight into prosand cons of various techniques and learn to adaptprocess activities to the context through practice,experience, and feedback.

In future we intend to increase the number of project context parameters or in other way it would be better to make it dynamic i.e., a configuration management utility should be included in the KEEREDECS. This utility will be heavily supported by the experience base by adding, updating and removing new project context parameters. Apart from that category definition, parameter domain values and much more will be performed using that module in next version of KEEREDECS.

We will also try to increase the capability of CBR module of KEEREDECS by including multiple Fuzzy Algorithms in it. It will help Requirements Engineer to choose one of Fuzzy Logic amongst many to get the better performance or at least compare the result produced by different logics and go for the best one as per his/her judgment/experience. Also different fuzzy membership function will be embedded in the KEEREDECS framework CBR so that requirement engineer should have choice to choose amongst multiples. This enhancement in the framework will be properly supported by KEEREDECS case base as it is already in normalized form.

We intend to increase the number of project context parameters or in other way it would be better to make it dynamic i.e., a configuration management utility should be included in the KEEREDECS. This utility will be heavily supported by the experience base by adding, updating and removing new project context parameters. Apart from that category definition, parameter domain values and much more will be performed using that module in next version of KEEREDECS.

As we know there has been a large number of RE techniques and historical datain the industry and in research areas. We have tried our best to collect as per maximum number of those techniques. But as we know our main focus was to design a framework so in future for much more better results a complete research activity will be conducted to remove any deficiency on this part.

Besides, we will improve the objective function system so that it suggests not only the RE techniques but also their sequence. We also intend to test the proposed framework in large scale industrial set-up. Also we have a plan to make a quality framework for KEEREDECS to check the proposed RE process quality.

References

[1] N.A.M. Maiden and G. Rugg, ACRE: selectingmethods for requirements acquisition, 1996

[2] Toshihiko Tsumaki and Tetsuo Tamai, AFramework for Matching Requirements EngineeringTechniques to Project Characteristics and SituationChanges, Proceedings of SREP'05, Paris, France, August 29–30, 2008, Eds. Ralyté J, Ågerfalk PJ andKraiem N

[3] Li Jiang, Armin Eberlein, Behrouz H. Far, A casestudy validation of a knowledge-based approach forthe selection of requirements engineering techniquesRequirementsEng. (2009) 13:117–146

[4] Damian D, Chisan J., An empirical study of the complex relationships between requirementsengineering processes and other processes that lead to payoffs in productivity, quality, and risk management. IEEE Trans SoftwEng 32(7):433–453, 2006

[5] Berry D, Damian D, Finkelstein A, Gause D, HallR, Wassyng A., To do or not to do: if the requirements engineering payoff is so good, why aren't more companies doing it?'', Proceeding of Requirements Engineering 2005.

[6] Damian D, Zowghi D, Vaidyanathasamy L, Pal Y., An industrial case study of immediate benefits of requirements engineering process improvement at the Australian Center for Unisys Software. Int J EmpirSoftwEng 9(1–2):45–75, 2004

[7] Hall T, Beecham S, Raner A (2002) Requirements problems in twelve software companies: an empirical analysis. IEE ProcSoftw149 (5)

[8] Jiang L (2005) A framework for requirements engineering process development. University of Calgary, PhD Thesis, Sept. 2005

[9] Lester O. Lobo and James D. Arthur, "Effective Requirements Generation: Synchronizing Early Verification& Validation, Methods and Method Selection Criteria", eprintarXiv:cs/0503004, 2005.

[10] Sarah Powell, Frank Keenan, Kevin McDaid,"Enhancing Agile Requirements Elicitation With Personas", IADIS International Journal on Computer Science and Information Systems, Vol. 2, No. 1, pp.82-95, ISSN: 1646-3692.

[11] Zheying Zhang, "Effective Requirements Development - A Comparison of Requirements

Elicitation techniques", SQM2007 conference.

[12] Sai Ganesh Gunda," Requirements Engineering: elicitation Techniques", Master's thesis software Engineering 2008, university West, Trollhattan.

[13] S. Kausar, S. Tariq, S. Riaz, A. Khanum, “Guidelines for Selecting Requirement ElicitationTechniques,”2010

[14] Glass, R.L.: The software-research crisis, Software, IEEE, 11(6), 42 – 47 (1999)

[15] Stallinger F., Grunbacher P., “System dynamicsmodelling and simulation of collaborative requirements engineering”, Journal of Systems and Software, Volume 59, Number 3, pp. 311-321(11), ,15 December 2001.

[16] Mannio M, Nikula U (2001), Requirements elicitation using a combination of prototypes and scenarios. Technical Report. Telecom Business Research Center, Lappeenranta University of Technology, P.O. Box 20, FIN-53851 Lappeenranta, Finland, Lappeenranta, 2001, ISBN 951-764-528-7

[17] Browne GJ, Ramesh V (2002) Improving Information requirements determination: a cognitive perspective. Inform Manage 39:625–645

[18] Kotonya, G., and I. Sommerville, Requirements Engineering, Wiley, 1998

[19] Glass, R., “Searching for the Holy Grail ofSoftware Engineering,” Comm. of the ACM, 45, 5(May 2012), pp. 15-16.

[20] Yadav, S., et al., “Comparison of Analysis Techniques for Information Requirements

Determination,” Communications of the ACM, 31, 9(Sept 1988).

[21] Hickey, A., and A. Davis, “A Tale of Two Ontologies: A Basis for Systems Analysis Technique Selection,” Americas Conference of Computer Information Systems (AMCIS), 2010.

[22] Li Jiang , Armin Eberlein , Behrouz H. Far ,Majid Mousavi, A methodology for the selection ofrequirements engineering techniques, SoftwSystModel (2008) 7:303–328, DOI 10.1007/s10270-0070055-y

[23] Zadeh, L. (1965), “Fuzzy sets”, Information and Control, 8: 338–353

[24] Matthew J Bickerton, Jawed Siddiqi, the Classification of Requirements Engineering Methods

[25] Hubert F. Hofmann, Franz Lehner,Requirements Engineering as a Success Factor in Software Projects, IEEE SOFTWARE July / August 2001, pp. 58-66

[26] An Objectives-Driven Process for Selecting Methods to Support Requirements Engineering Activities Lester O. Lobo and James D. Arthur, Software Engineering Workshop, 2009. 29th AnnualIEEE/NASA,on page(s): 118 – 130

[27] AWRE 2012, Requirements Engineering Process Models in Practice Sacha Martin1, AybükeAurum1, Ross Jeffery2, Barbara Paech3, 141-155

[28] Jiang, L., Eberlein, A. (2013). Requirements Engineering: A Review and a Proposal. In Proceedings of the Third Alberta Software Engineering Research Consortium (ASERC)Workshop on Quantitative and Soft-Computing Based Software Engineering (QSSE2003), Banff, Alberta, Canada.

[29] Jiang, L., Eberlein, A., & Far, B.H. (2004): Amethodology for Requirement Engineering process development. In ECBS 11th IEEE International Conference on the Engineering of Computer-Based Systems, Brno, Czech Republic.

[30] Pfeiffer, S., &Eberlein, A. (2002).Requirements engineering for dynamic markets. InSoftware Engineering and Applications, 8(1), 374 –381

[31] Important factors in selecting Requirements Engineering Techniques EsmaeilKheirkhah, AzizDeraman

[32] Requirements for Requirements Engineering Techniques Linda Macaulay, Proceedings of ICRE'96, pp. 157-164

[33] Requirements Elicitation and Elicitation Technique Selection: A Model for Two KnowledgeIntensiveSoftware Development Processes, Ann M.Hickey Alan M. Davis, Proceedings of the 36thHawaii International Conference on System Sciences (HICSS'03),

[34] Macaulay, L. A. (1996): Requirements Engineering, Springer-Verlag

[35] P. Loucopoulos, V. Karakostas, System Requirements Engineering, 2005

[36] Aamodt, A., Plaza, E, "Case-Based Reasoning: Foundational Issues, Methodological Variations and System Approaches", AI communications (1994), pp.35-39.

[37] A. Khanum, M.Mufti, "Fuzzy Case Based Reasoning for Facial Expression Recognition," FuzzySets and Systems (Elsevier), 2009

[38] UfukCebeci, "Fuzzy AHP-based decision support system for selecting ERP systems in textile industry by using balanced scorecard," Expert Systems with Applications, 36(2009), 8900-8909

[39] K.J.Hammond. Case-based planning: viewing planning as a memory task. Academic Press Professional, Inc., San Diego, CA, USA, 1989.

[40] Kristian J. Hammond, Case-Based Planning: A Framework for Planning from Experience, COGNITIVE SCIENCE 14, 385-443 (1990)

[41] H. Aris, and S. S. Salim, The Development of a Simplified Process Model for CBSD, The International Arab Journal of Information Technology, Vol. 4, No. 2, pp. 89-96, April 2007

[42] A case study validation of a knowledge-based approach for the selection of requirements engineering techniques Jiang Æ Armin Eberlein Æ Behrouz H. Far.

[43] TeknomoK. (2004) Similarity Measurement,http://people.revoledu.com/kardi/tutorial/Similarity/, 2004

[44] Brian S. Everitt, S. Landau and M. Leese (2001) Cluster Analysis, Fourth Edition, Oxford University Press Inc.

[45] Winston P. H., Artificial intelligence, 3rd (repr. with corrections 1993) ed. Reading, Mass.: Addison-Wesley, ISBN: 0-201-53377-4, 1993.

[46] Althoff K.-D. and Nick M. M., "How to Support Experience Management with Evaluation - Foundations, Evaluation Methods, and Examples for Case-Based Reasoning and Experience Factory." Berlin: Springer Verlag, 2012 (forthcoming, Accepted for LNAI series).

[47] Birk A., Surmann D., and Althoff K. D., "Applications of knowledge acquisition in experimental software engineering," presented at 11th European Knowledge Acquisition Workshop (EKAW'): Knowledge Acquisition, Modeling, and Management, Berlin, Germany, 2009.

[48] Brandt M. and Nick M., "Computer-supported reuse of project management experience with an experience base," presented at Third International Workshop on Advances in Learning Software Organizations (LSO 2001), Berlin, Germany, 2008

[49] Dick S. H., Computational Intelligence in Software Quality Assurance. PhD Thesis. Department of Computer Science and Engineering, College of Engineering, University of South Florida, 2002.

[50] Khoshgoftaar T. M., Software engineering with computational intelligence, vol. 731. Boston: Kluwer Academic Publishers, ISBN: 1-402-07427-1, 2003.

[51] Martin Shepperd, Case-based Reasoning and Software Engineering

[52] Aamodt, A. and E. Plaza, Case-based reasoning: foundational issues, methodical variations and system approaches. AI Communications 7(1), 2004.

[53] Little, R.J.A. and D.B. Rubin, Statistical Analysis with Missing Data. 2nd ed, New York: John Wiley & Sons, 2002.

[54] Maiden, N.A., Analogy as a paradigm for specification reuse. Software Engineering Journal 6(1): pp3-15, 2001.

[55] Maiden, N.A. and A.G. Sutcliffe, Exploiting reusable specifications through analogy. Communications of the ACM 35(4): pp55-64, 1992.

[56] Bisio, R. and F. Malabocchia. Cost estimation of software projects through case base reasoning. 1st Intl. Conf. on Case-Based Reasoning Research & Development. pp11-22: Springer-Verlag, 1995.

[57] Schenker, D.F. and T.M. Khoshgoftaar. The application of fuzzy enhanced case-based reasoning for identifying fault-prone modules. 3rd IEEE International High-Assurance Systems Engineering Symposium. Pp90-97, Washington, D.C.: IEEE Computer Society, 1998.

[58] L.R. Medsker, Hybrid Intelligent Systems, Kluwer Academic Publishers, 1995 (second printing 1998).

[59] Beatrice Alenljung. Decision-making in the Requirements Engineering Process: A Human-centred Approach, Linköping University Licentiate of Philosophy Thesis, December 2005

[60] Ebert, C. & Wieringa, R.J. (2005) Requirements engineering: Solutions and trends. In: A. Aurum & C. Wohlin (Eds.) Engineering and managing software requirements (pp 453-476). Berlin, Germany: Springer

[61] Beatrice Alenljung. Envisioning a Future Decision Support System for Requirements Engineering – A Holistic and Human-centred Perspective, Linköping University Dissertation, 2008

[62] Aurum, A. & Wohlin, C. (2003) the fundamental nature of requirements engineering activities as a decision-making process. Information and Software Technology, 45, 945-954

[63] Orasanu, J. & Connolly, T. (1993) the reinvention of decision making. In: G.A. Klein, J. Orasanu, R. Calderwood & C.E. Zsambok (Eds.) Decision making in action: Models and methods (pp 3-20). Norwood, New Jersey: Ablex Publishing

[64] Ngo-The, A. & Ruhe, G. (2005) Decision support in requirements engineering. In: A. Aurum & C. Wohlin (Eds), Engineering and managing software requirements (pp 267-286). Berlin, Germany: Springer

[65] Li Jiang, Armin Eberlein. Analysis of Requirements Engineering Techniques Using Clustering

[66] Jiang L, Eberlein A, Far BH, Majid M (2009) An Innovative Requirement Engineering Technique Selection Model, Technical Reports, August, 30, 2010, Dept. of Electrical and Engineering, University of Calgary, http://www.enel.ucalgary.ca/~jiangl/research/ indexOfResearch.htm

[67] Li Jiang, A Framework for Requirements Engineering Process Development, PhD Thesis, University of Calgary, Sep. 2006

[68] Simon Haykin, Neural Networks: A Comprehensive Foundation, 2ndEd., Pearson Prentice Hall, 2007

[69] John R. Koza, Genetic Programming, MIT Press, 2011

[70] Didier Dubois, Walenty Ostasiewicz, Henri Prade. FUZZY SETS: HISTORY AND BASIC NOTIONS, In Fundamentals of Fuzzy Sets.

[71] Zsolt Csaba JOHANYAK, Kalman Milan, BOLLA, DEVELOPMENT OF A TOOLBOX SUPPORTING FUZZY CALCULATIONS, International Symposium on Advanced Engineering and Applied Management, pp. 155-158, 2010

[72] Bjarne K. Hansen, ANALOG FORCASTING OF CEILING AND VISIBILITY USING FUZZY SETS, 2nd Conference on Artificial Intelligence, American Meterological Society, pp, 1-7, 2000

APPENDIX – A: Survey Form

Participants Information

Personal Information	
Name:	
Designation:	
Qualification:	
Experience(in years)	
Contact No:	
Organization Information	
Organization Name	
No. of Employees	
Type(Public/Private)	
Organization Age	

S#	**Questions**	**Answer**				
		Strongly Agree	**Agree**	**Medium**	**Disagree**	**Strongly Disagree**
Category-I : Administration						
1	The RE techniques recommended are appropriate for PROJECT-II project					
2	The recommended techniques was up to the mark for reduction of re-work during all phases of software development life cycle					
3	The recommended techniques was good enough to complete the project within intended time span.					
4	The final version of SRS was of high satisfaction for the both the customer and Administration					
5	The overall organization of Requirements Engineering process for PROJECT-II was much better than PROJECT-I and later projects					
Category -II: **Coders**						
6	Both Functional and Non Functional requirements was easily mapped to coding					
7	The overall rate of change of requirements					

	during coding phase was much less than PROJECT-I and other projects done before					
8	The SRS document was precise and understandable than before					
9	RE Techniques used for requirements specification in SRS was easy to understand and was within acceptable range					
10	It was easy to trace back the requirements					
11	The recommended techniques was good enough to complete the project within intended time span					
12	The overall quality of the documentation is high					
Category -III: Requirements Engineers, Analyst and Designer						
13	The main issue of the PROJECT-II project was easily addressable by the recommended techniques					
14	Requirements Modeling in SRS was easily mapped to designing/modeling constructs					
15	Usage of Modeling notation language such as unified modeling language was helpful and time saving in mapping requirements to modeling constructs					
16	Inclusion of more formal RE techniques in recommended techniques would increase the quality of SRS and modeling					
17	With appropriate training struggle and cost to newly recommended techniques can be easily overcome which were never used in previous project by keeping cost and difficulty factor					
18	Requirements Engineering process was much more structured and planned than before					
19	Recommendation done by KEEREDECS framework was good enough for project type like PROJECT-II					

APPENDIX – B: Project Context Parameters

Category	Attribute	Units
Continues	**Scope(S)**	If RV >800And RV <=1000 Then S = Very Big If RV > 600 And RV <=800 Then S = Big If RV >400 And RV <=600 Then S = Medium If RV > 200 And RV <=400 Then S = Small If RV >0 And RV <= 200 Then S = Very Small **** RV = Requirements Volume***
Ordinal	**Complexity**	**{Very high,High,Medium,Low,Very Low}**
Continues	**Requirements Instability(RI)**	If CD >80% And CD <= 100% Then RI = Very High If CD > 60% And CD <= 80% Then RI = High If CD >40% And CD <= 60% Then RI = Medium If CD > 20% And CD <= 40% Then RI = Low If CD >0% And CD <=20% Then RI = Very Low **** CD = Change Degree***
Ordinal	**Degree of Safety Criticality**	**{Very high,High,Medium,Low,Very Low}**
Ordinal	**Product Quality Criteria**	**{Very high,High,Medium,Low,Very Low}**
Ordinal	**Time Constraints**	**{Very high,High,Medium,Low,Very Low}**
Ordinal	**Cost Constraints**	**{Very high,High,Medium,Low,Very Low}**
Continues	**Team Size(TS)**	If TM >80And TM <=100 Then TS = Very Big If TM >60 And TM <= 80 Then TS = Big If TM >40 And TM <= 60 Then TS = Medium If TM > 20 And TM <= 40 Then TS = Small If TM > 0 And TM <= 20 Then TS = Very Small **** TM = Team Members***
Ordinal	**Domain Acquaintance**	**{Very high,High,Medium,Low,Very Low}**
Ordinal	**Team RE Knowledge**	**{Very high,High,Medium,Low,Very Low}**
Ordinal	**Degree of Requirements**	**{Very high,High,Medium,Low,Very Low}**

	Knowledge	
Ordinal	**Skilled Facilitator Availability**	**{Very high,High,Medium,Low,Very Low}**
Ordinal	**Stakeholder Heterogeneity**	**{Very high,High,Medium,Low,Very Low}**
Continues	**Innovation Degree(ID)**	If NF >80And NF <=100 Then ID = Very High If NF >60 And NF <= 80 Then ID = High If NF >40 And NF <= 60 Then ID = Medium If NF > 20 And NF <= 40 Then ID = Low If NF > 0 And NF <= 20 Then ID = Very Low **_* NF = New Features_**
Ordinal	**Customer Availability**	**{Very high,High,Medium,Low,Very Low}**
Ordinal	**Degree of Reusability**	**{Very high,High,Medium,Low,Very Low}**
Ordinal	**Degree of Implicit Knowledge**	**{Very high,High,Medium,Low,Very Low}**
Ordinal	**Degree of Outsourcing**	**{Very high,High,Medium,Low,Very Low}**
Nominal	**Business Client Association**	**RFQ :** Request for Quotation **PCD :** Particular Client Demand **GPM :** General Product for Marketplace **PT :** Product Tailoring **WO :** Business Needs within Same Organization
Nominal	**Project Category**	Organic Category, Semi-detached, Embedded Hardware/Software Systems, Communication and distributed system, System software development
Nominal	**Product Type**	New, Upgraded, Serial, COTS

APPENDIX – C: Phase Wise CI Formulas

- **Functionally Equivalent**

$$\boldsymbol{TSe} = \frac{\sum_{j=1}^{7} \boldsymbol{T(Aj) - T'(Aj)}}{\mathbf{7}} <= \ \boldsymbol{\alpha}$$

$$\boldsymbol{TSa} = \frac{\sum_{j=8}^{15} \boldsymbol{T(Aj) - T'(Aj)}}{\mathbf{8}} <= \ \boldsymbol{\alpha}$$

$$\boldsymbol{TSv} = \frac{\sum_{j=16}^{23} \boldsymbol{T(Aj) - T'(Aj)}}{\mathbf{8}} <= \ \boldsymbol{\alpha}$$

$$\boldsymbol{TSd} = \frac{\sum_{j=24}^{26} \boldsymbol{T(Aj) - T'(Aj)}}{\mathbf{3}} <= \ \boldsymbol{\alpha}$$

- **Functionally Opposite**

$$\boldsymbol{TDe} = \ < \frac{\sum_{j=1}^{7} \boldsymbol{T(Aj) - T'(Aj)}}{\mathbf{7}} <= \ \beta$$

$$\boldsymbol{TDa} = \ < \frac{\sum_{j=8}^{15} \boldsymbol{T(Aj) - T'(Aj)}}{\mathbf{7}} <= \ \beta$$

$$\boldsymbol{TDv} = \ < \frac{\sum_{j=16}^{23} \boldsymbol{T(Aj) - T'(Aj)}}{\mathbf{7}} <= \ \beta$$

$$\boldsymbol{TDd} = \ < \frac{\sum_{j=24}^{26} \boldsymbol{T(Aj) - T'(Aj)}}{\mathbf{7}} <= \ \beta$$

APPENDIX – D: Phase Wise Objective Functions

- **Elicitation Phase**

$$\boldsymbol{Oe} = \sum_{\boldsymbol{t=1}}^{\boldsymbol{n}} (\textstyle\sum_{\boldsymbol{j=1}}^{\boldsymbol{7}} \boldsymbol{T}(\boldsymbol{Aj}) * \boldsymbol{Bj})$$

- **Analysis Phase**

$$\boldsymbol{Oa} = \sum_{\boldsymbol{t=1}}^{\boldsymbol{n}} (\textstyle\sum_{\boldsymbol{j=8}}^{\boldsymbol{15}} \boldsymbol{T}(\boldsymbol{Aj}) * \boldsymbol{Bj})$$

- **Documentation Phase**

$$\boldsymbol{Od} = \sum_{\boldsymbol{t=1}}^{\boldsymbol{n}} (\textstyle\sum_{\boldsymbol{j=16}}^{\boldsymbol{23}} \boldsymbol{T}(\boldsymbol{Aj}) * \boldsymbol{Bj})$$

- **V & V Phase**

$$\boldsymbol{Ov} = \sum_{\boldsymbol{t=1}}^{\boldsymbol{n}} (\textstyle\sum_{\boldsymbol{j=24}}^{\boldsymbol{26}} \boldsymbol{T}(\boldsymbol{Aj}) * \boldsymbol{Bj})$$

APPENDIX – E: Process Context Parameters

Parameter Name	Parameter Type
Facilitate Communication	Elicitation Phase
Understand Social Issues	//
Get Domain Knowledge	//
Get Implicit Knowledge	//
Identify Stakeholders	//
Identify NFR	//
Identify Viewpoints	//
Modeling Requirements	Analysis Phase
Help Analyze And UnderstandRequirements	//
Help Analyze And Understand NFR	//
Facilitate Negotiation With Customers	//
Prioritize Requirements	//
Identify Accessibility Of The System	//
Model Interface Requirements	//
Identify Reusable Requirements	//
Represent Requirements	Documentation Phase
Verify Requirements Automatically	//
Completeness Of The Semantics Of The Notation	//
Write Unambiguous And Precise Requirements	//
Write Complete Requirements	//
Management Of Requirements	//
Modularity	//
Implementability	//
Identify Ambiguous Requirements	Verification & Validation Phase
Identify Interactions (inconsistency, conflict)	//
Identify Incomplete Requirements	//