Requirement Engineering in Component Selection

Thesis submitted in partial fulfillment of the requirements for the award of degree of

Master of Engineering

In

Software Engineering

By:
Rashmi Gupta

(800831009)

Under the supervision of:

Mrs. Shivani Goel

Assistant Professor

COMPUTER SCIENCE AND ENGINEERING DEPARTMENT

THAPAR UNIVERSITY

PATIALA – 147004

June 2010

Certificate

I hereby certify that the work which is being presented in the thesis entitled, **"Requirement Engineering in Component Selection"**, in partial fulfilment of the requirements for the award of degree of Master of Engineering in Software Engineering submitted in Computer Science and Engineering Department of Thapar University, Patiala, is an authentic record of my own work carried out under the supervision of *Mrs. Shivani Goel* and refers other researcher's works which are duly listed in the reference section.

The matter presented in this thesis has not been submitted for the award of any other degree of this or any other university.

Rashmi

(Rashmi Gupta)

This is to certify that the above statement made by the candidate is correct and true to the best of my knowledge.

Shivani Goel.

(Mrs. Shivani Goel)

Computer Science and Engineering Department

Thapar University
Patiala

Countersigned by

(RAJESH BHATIA) 6/6/10

Head

Computer Science & Engineering, Department

(R.K.SHARMA) 6/6/10

Dean (Academic Affairs)

Thapar University.

Acknowledgement

First of all, I would like to thank to GOD and my parents for giving me the strength and health to do this thesis work.

I wish to express my deep gratitude to Mrs. Shivani Goel, Assistant Professor, *Computer Science and Engineering Department*, TU, Patiala for providing her guidance and support throughout this thesis work.

I am thankful to Dr. Rajesh K. Bhatia, Head, *Computer Science & Engineering Department*, TU, Patiala, for the motivation and inspiration towards the practical implementation of the thesis work. I would like to thank all the other faculty members & staff members who were always there at the need of time and provide with all the help & support, which I required for the completion of the thesis.

Last but not least, I am thankful to all my friends Sugandha, Shirin, Sushila, Yogesh and Tikka for encouraging and providing me useful information during my work.

Finally, my special thanks go to authors whose work I have consulted and quoted in this thesis.

Rashmi

Rashmi Gupta

800831009

ME (Software Engineering)

Computer Science & Engineering Department

Thapar University, Patiala

Abstract

In the COTS selection process, it is unavoidable to encounter mismatches between system requirements and COTS products' capabilities. These mismatches are the result of the lacking COTS attributes or their overmatch with the system requirements.

Many researchers have proposed various techniques to search and select COTS components. But only few have worked upon the handling of mismatches. These techniques have their own advantages and disadvantages.

This thesis work proposes a COTS selection approach that aims at addressing COTS mismatches among COTS attributes and system requirements during the requirement engineering phase. In this process COTS evaluation, a core activity of COTS selection, is performed using the Goal Question Metrics (GQM) based Feature Analysis technique. GQM based Feature Analysis is an evaluation strategy used for measuring the compliance of the COTS capabilities with desired features using the formulated metrics. A case-study is used to illustrate this process and to discuss its feasibility.

Keywords: Goal Question Metric paradigm, Feature Analysis, COTS evaluation & selection.

Abbreviations

AHP	Analytic Hierarchy Process
BIN	Binary
CARE	COTS Aware Requirement Engineering
CBD	COTS Based Development
COTS	Commercial Off The Shelf
CRE	COTS Requirement Engineering
DSS	Decision Support System
GQM	Goal Question Metric
IDS-SA	Interactive Decision Support-Sensitivity Analysis
MCDM	Multi Criteria Decision Making
MiHOS	Mismatch Handling aware cOts Selection
MV	Mismatch Value
NFR	Non-Functional Requirement
ORDN	Ordinal
OTSO	Off The Shelf Option
PORE	Procurement Oriented Requirement Engineering
Satf	Satisfaction Function
SUD	System Under Development
WSM	Weighted Score Method
WYSIWYG	What You See Is What You Get

Table of Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Background

In the computer market, existing technologies get obsolete speedily than in any other industry. This rapid-fire pace demands continuous increase in productivity and competitiveness for the computer market. Many industries like health care, manufacturing, transportation and retail – depend on computer products and services for their vitality. So, in order to deliver the products and services to these industries, product development life-cycles are continually decreasing.

In the last several years, software development has been upraised by reusing the existing software or the commercial-off-the shelf (COTS) software products. So, the purpose of software reuse is to develop large systems by incorporating previously developed or existing software. By this way, software development organisation is able to cut down development time, costs and effort required for the development of the same from the scratch. While developing the software with reuse, existing potential reusable software are need to be evaluated whether they meet the application-specific requirements or not. If not, then it requires the modification in the application-specific requirements or adaptation in COTS itself. After this, compatibility of this reusable software is checked with the desired environment. In this way, the idea of reuse-centred software development is known as **Reuse-Based Development**.

1.2 Reuse Based Development

The main idea of the reuse-based development is building systems from pre-existing components. This assumption has several outcomes for the system development lifecycle. First, the concept of Development with Reuse is entirely different from Development for Reuse. In the development with reuse methodology, the reusable components should already been developed and possibly used in other products when the system development process starts. But in Development for Reuse, components are developed for general purpose and reused in many applications that may or may not exist presently. Also, requirements and business ideas are not much application specific. Second, a new separate process will appear: Exploration and selection of the

components are the additional tasks that demand much consideration in the development with reuse. Reuse Based Development is focused on the identification of reusable entities and relations between them, beginning from the system requirements and from the availability of already existing components. The difference between these two methodologies is well discussed through the V-model [1] given below.

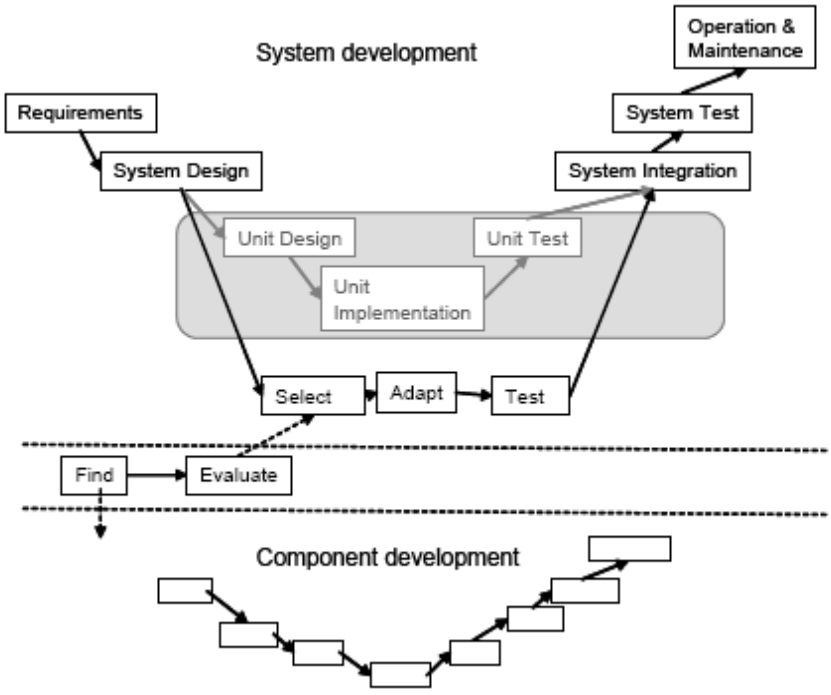

Figure 1.1: V-Model for Reuse Based Development [1]

Instead of performing activities that often are time and efforts consuming like unit design and implementation, simply selection and evaluation of the appropriate components is performed and then integrated in the system. However, two problems appear here which break this simplicity: (i) the availability of the component for the desired application, and (ii) the fitness of the selected components to our overall design [1]. The first fact shows that we must have a process for finding components. This process includes activities for finding the components, and then the component evaluation. The second fact indicates for a need of component adoption and testing before it can be integrated into the system. But the process of finding, evaluation and selection of the components need much consideration as it comes early in the cycle.

1.3 Need of Reuse Based Development

In today's world of competition, more and more companies compete in order to deliver the product faster and provide better innovative services to their customers. As a result of faster delivery, software organizations are not able to produce manageable, high quality, cost-effective software. This results in decreased business performance of the organisation. In order to improve the business performance, the software companies have to shorten the time required to bring a product to the market, reducing software development and maintenance costs, and increasing the quality of their software. To meet these business oriented goals, organizations are switched to reuse based development. Following are some of the benefits that reuse based development provide [2]:

- Increased Product Trustworthiness as the reusable component are tried and tested in working environments.
- High Maturity as COTS products go through multiple releases starting from beta versions to final versions.
- It reduces the risks involved in the development process as the cost of development is lesser as compared to traditional development.
- It reduces the risk of failure as the components are certified for their performance, quality and reliability.

1.4 Reusable Software Product

Different researchers have attempted to define what we mean by a "reusable software product". In this thesis, we adopt the definition provided by Vigder et al. [3] which defines a reusable software product or COTS software product as:

"A reusable software product: developed by a third party (who controls its ongoing support and evolution); bought, licensed, or acquired for the purposes of integration into a larger system as an integral part, i.e. that will be delivered as part of the system to the customer of that system; which might or might not allow modification at the source code level, but may include mechanisms for customization; and is bought and used by a significant number of systems developers."

Typical examples of COTS products are Notepad, Word Processors, Picture Manager, Internet Explorer provided by Microsoft, Adobe Photoshop etc.

1.5 Challenges to Software Reuse

The use of COTS software products provides various benefits stated in section 1.3. Besides the benefits several challenges are also encountered during the software development with reuse. Many system failure occurred in past because the system failed to overcome these challenges. For example, the London Ambulance Service fiasco in 1992, in which the system descended into chaos and reverted back to manual operation partly because of inappropriate COTS selection which would be discussed in section 1.6 ([4],[5]). Following are the challenges to Reuse Based Development paradigm [2]:-

- *Requirement Engineering and COTS Selection*

  Some components don't suit exactly with the desired requirements. So, an analysis of the feasibility of requirements in relation to the available components is done which is followed by the reformulation of requirements or component adaptation sometimes.

- *Lack of Tool Support*

  Purpose of software reuse is to build systems from reusable components simply and efficiently and to deliver the product in due time, and this can only be achieved with the help of tool support. For example- tool for component selection and retrieval, component test, tool for managing the component repositories.

- *COTS Integration*

  In Intensive COTS based systems, multiple COTS are selected. As a result, their interoperability issues arises which puts a challenge to COTS integration.

1.6 COTS Selection

Building software systems from various commercial off-the-shelf (COTS), software components are very appealing because they can reduce the development risks and costs while increasing the functionality and capability of software systems. Selecting the components with the desired capabilities and lesser mismatches with the stakeholder's requirements is of much concern. The candidate components are gone under the evaluation phase which is the most crucial task of the selection cycle and resulting in various mismatches. Many COTS selection techniques are already

proposed, but only few, like PORE, MiHOS, has focussed the concept of mismatch handling, which resolve mismatches at prior. This reduces the time and effort spent over the integration of COTS in COTS based development.

1.6.1 COTS Selection Criteria

Following are some of the criteria on the basis of, COTS selection should be performed ([2], [6]):

- *Acquisition of Changing Requirements (C1)*

  Requirements form main criteria for understanding the system under development. This is because it is requirements that the concerned stakeholders have to communicate about, agree/disagree upon. They become criteria for evaluating and selecting of candidate components and are embedded in the legal contract (SRS). Requirements even provide acceptance criteria to check when the system is delivered that it meets customer's expectations. Most current methods and tools support systems design and integration but neglect the requirements engineering and product selection processes that must precede design and integration. In spite of this lack of focus on requirements engineering, there is an emerging concept of iterative requirements engineering process that entertains the requirement change in component based development which leads to greater customer satisfaction.

- *Multiple Stakeholders and the Interaction (C2)*

  The COTS selection process usually involves multiple stakeholders, each with his/her set of needs, preferences, and constraints, which may be in conflict with those of other stakeholders, thus leading to a negotiation problem. Moreover, the stakeholders have to establish a means for information sharing and for group representation of the problem in order to select COTS that satisfies the needs of every stakeholder as much as possible.

- *Presence of Similar COTS Components (C3)*

  When COTS are procured from multiple vendors there arise a quite possible situation when two or more searched COTS components have the same

functionalities but differ only in the name & their quality aspects. This leads to extra overhead while evaluating their characteristics.

- *Non-Functional Requirement (NFR) Description (C4)*

  A complete COTS selection process includes considering the desired functionality together with the understanding of non-functional aspects (performance, reliability, scalability, user friendly...). In general, vendors don't provide the non-functional requirement description. So, it emerges out as a challenging activity to take out a component with the required features and "lities" as well.

- *Decision Making Analysis (C5)*

  COTS selection is not based on single decision; it has to pass over a level of decisions that makes a hierarchical process. The hierarchical process comprises of the search for candidate components, evaluation of candidate list, and compliance of the evaluated components with non-functional aspects and so on. There are available many decision making techniques which facilitates DSS like Multi Criteria Decision Making (MCDM), game theory etc [7].

- *Cost and Benefit Analysis (C6)*

  From the alternative list of COTS components for a desired functionality, cost and benefit analysis of the component in regard of the organisation forms an elimination criteria. Cost estimation for the alternative components comprise of acquisition cost, adaptation cost and integration cost in the owner's environment

- *Handling Mismatches during Component Selection (C7)*

  In CBD, in order to decrease the development time, integration and testing time increases. There are various possibilities of mismatches, for example: (1) COTS products have extra capabilities that are not desired by stakeholders, and (2) COTS products don't meet all specified requirements. So, in order to decrease the possibility of these kind of mismatches & integration time,

integration issues (architectural & COTS mismatches) should be considered while the COTS selection.

1.6.2 COTS Evaluation

COTS evaluation is one of the core activities of COTS selection. There are a number of different COTS evaluation methods, each with its own features; some place emphasis on different quality attributes and some goal graph based evaluation, others use strong evaluation criteria definition. Based on these aspects, this thesis focuses on a Features Analysis [8] of the COTS product using Goal Question Metric (GQM) paradigm [9] applied to a single case study of Source Code Editor. This Features Analysis makes two contributions; on the one other hand it pinpoints some of the aspects that must be present in a COTS product and, on the other, it identifies the characteristics of the COTS evaluated. These strengths are determined by their dynamic behaviour in response to the features proposed. Proposed features are based on the GQM technique. Once the features and sub features are identified, metric should be defined in order for the evaluation of the COTS products and to get the satisfaction value of the different COTS. A metric in which the degree of satisfaction of the COTS must be measured or judged is defined.

1.7 Summary

In this chapter concepts related to Reuse-Based development and reusable software component are described. Also various benefits and challenges for software reuse are explored. Brief of the GQM based Feature Analysis for the COTS evaluation is also given in this chapter. Various COTS selection criteria are explored here but techniques would be reviewed in the next chapter.

1.8 Thesis Outline

In the thesis, Chapter 2 gives the overview of various COTS selection techniques and their drawbacks. Various decision making techniques for the evaluation purpose and Mismatch taxonomies, proposed earlier, would be reviewed also.

Chapter 3 includes the problem definition and the scope of the thesis work.

Chapter 4 includes proposed model for COTS selection. Proposed model is explained with the help of Data Flow Diagram for various phases and sub-phases.

Chapter 5 includes the validation of the proposed model with the help of the case study of Source Code Editor.

Chapter 6 includes the Conclusion and Future Scope.

Chapter 2

Literature Review

As discussed the various COTS selection criteria (C1-C7) in the previous chapter, selecting the components with the desired capabilities and maximum satisfaction with these criteria is of much concern. So, in this chapter different types of existing COTS selection techniques are investigated in subsections 2.1 to 2.5. Only few, like CARE, PORE, and MiHOS have focussed over the concept of mismatch handling (C7) at prior. This reduces the time and effort spent over the integration of COTS in COTS based systems. OTSO has considered the criteria Cost & Benefit analysis (C6) which estimates the cost & benefit for the developing organisation. Decision making techniques are also explained in the subsection 2.6.

2.1 Procurement Oriented Requirement Engineering (PORE)

2.1.1 Description of PORE

The PORE approach has three main components [10]:

- A Process Model that identifies undertaken fundamental processes like requirement acquisition/ validation, supplier selection, software package selection, package acceptance and management of system procurement.
- A Method Box that includes methods, techniques and tools that is available to help and achieve each of the processes.
- A Product Model that enables effective product evaluation and selection using the use case modelling, goals-based requirement methods and architecture modelling techniques.

PORE, [10] which support the requirements engineering and product evaluation/selection processes for CBSE development process. PORE uses an iterative process of requirements acquisition and product evaluation/selection as its main approach as discussed by Fox et al. in [11] and in figure 2.1.

- It selects products by rejection [12], (i.e. the products that do not meet core customer requirements are selectively and iteratively rejected and removed from the candidate list).

- It makes use of various tools and techniques like card sorting & laddering for requirement acquisition, MCDM [7] & AHP [13] for decision making during component selection.
- It makes use of the Feature Analysis technique [8] for scoring the match between COTS features and desired requirements.

2.1.2 Drawbacks of PORE

- Iterative acquisition of requirements is a tedious and complex task and thus time & effort consuming.
- It partially addresses the mismatch problem. As it doesn't talk of the influence of the mismatch on System Under Development (SUD) and how to handle that mismatch.
- It doesn't support the resolution of the issue regarding the similar COTS components acquired from multiple stakeholders.
- Cost & Benefit analysis of the procured COTS is not available.
- Drawbacks of AHP limit its use.

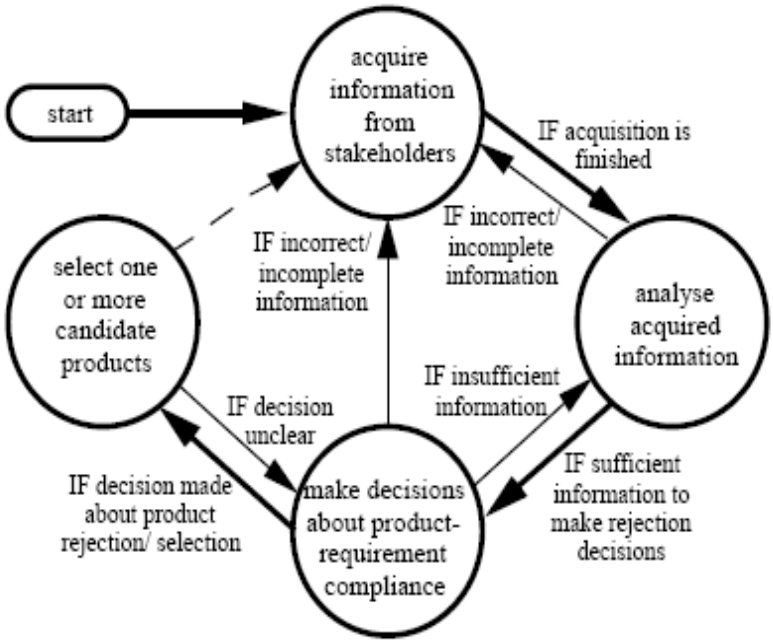

Figure 2.1: COTS Selection Process using PORE [17]

2.2 COTS Aware Requirement Engineering (CARE)

2.2.1 Description of CARE

The CARE approach is characterized as goal-oriented, agent oriented, knowledge based, and has a defined methodology or process. The Care process framework ([14], [15], [16], [17]) is based on five activities (as in figure 2.2):
Define Goals, Match Goals, Rank Components, Negotiate Changes and Select Components.

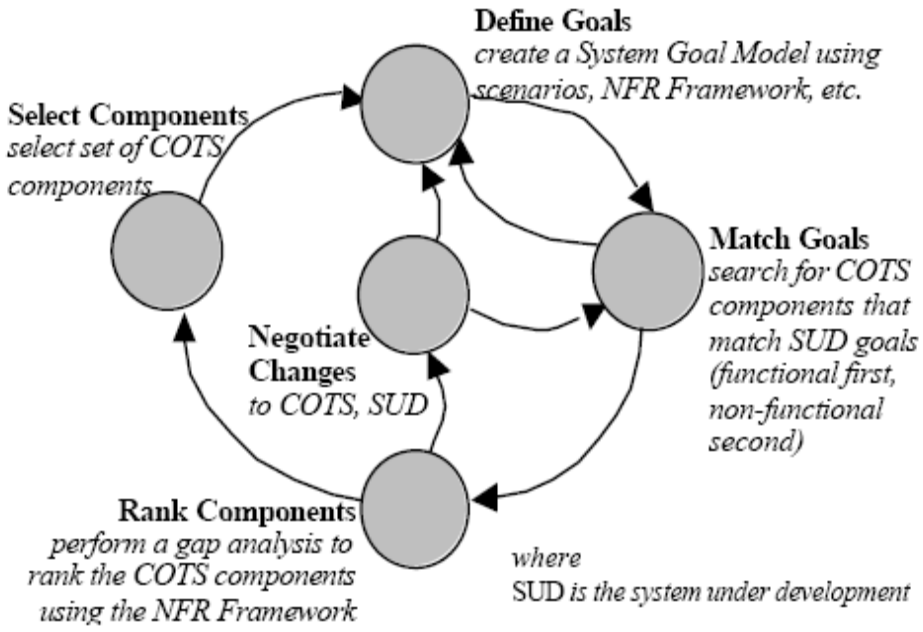

Figure 2.2: CARE- An iterative process for Component Based Development [16]

- It defines two types of requirement: Native (acquired from the stakeholders) and Foreign (acquired from the COTS capabilities).
- Knowledge Based: Because without the knowledge of the COTS components, they cannot be used [17]. Repository is maintained for storing the native requirements and the foreign requirements as well.
- Agent Oriented: The agents [18] are the various stakeholders involved in the SUD. An agent possesses characteristics including goals, beliefs and abilities that can make autonomous decisions, choices, and depends on other agents to accomplish goals, complete tasks, or furnish resources.

- Goal Oriented: The goals [19] are used to drive the development of the system. First, the goals are iteratively refined into system requirements which further refined into software, hardware, or interface requirements.
- It follows a well defined development process using COTS, so, it produces sustainable software. It is an iterative process for requirement acquisition, so it can welcome the changed objectives also.
- Maintenance of repositories, for knowledge based feature, of vendor's information that eases out the vendor selection problem.
- Use of NFR framework [20] leads to heavy documentation of goals given by the NFR. But it increases the efficiency of component selection for same goals in near future.
- It uses the elimination method by ranking the components based on the level of satisfaction of the requirements by the particular component.

2.2.2 Drawbacks of CARE

- Iterative nature of the elicitation, analysis, validation of requirements increases the time to deliver the frequent deliverables.
- Assumption of the characteristics of COTS components present in the repository puts restriction over the search of components for the desired goals of SUD.
- It partially addresses the mismatch problem as it doesn't talk of the influence of the mismatch on SUD and how to handle that mismatch.
- Drawbacks of AHP limit its use.

2.3 COTS Requirement Engineering (CRE)

2.3.1 Description of CRE

- CRE, as proposed by Alves et al. [21], is an iterative process for COTS component selection which aids the entertainment of changed requirements.
- It includes mainly 4 phases which goes iteratively.
    - Requirement acquisition
    - Product Identification

- Product Description
- Product Acceptance

- It is goal oriented, i.e. each phase is oriented by predefined goals. Each phase has a template (in figure 2.3) that includes some guidelines and techniques for requirements acquisition/modelling and products evaluation. These templates describe the main goals as well as the final results of each phase.

Template1
Goals:
Final result:
Information and requirements to acquire:
Steps sequence:
Important considerations:

Figure2.3: Part of CRE Templates [21]

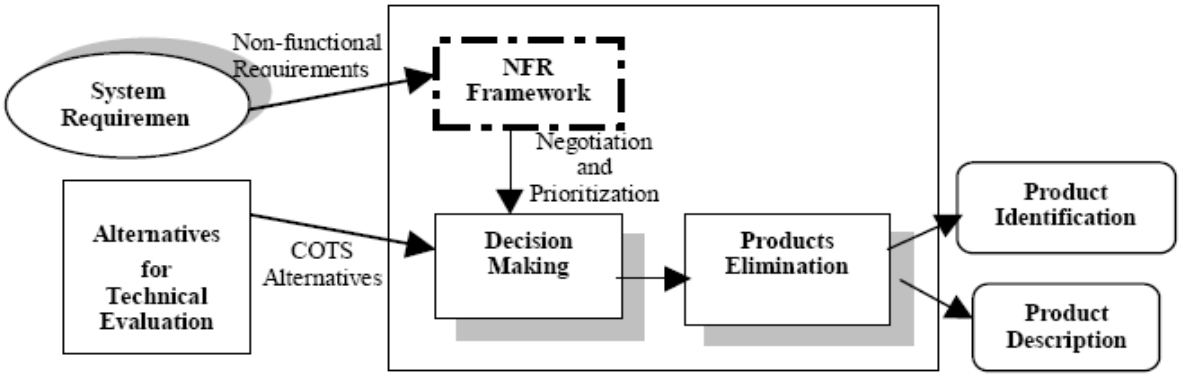

Figure 2.4: NFR Framework in Requirement Acquisition [22]

- NFR framework [20] is focussed as the main evaluation criteria for COTS as explained in figure 2.4 and thus selection of COTS component is based on rejection on the basis of time restriction, cost & benefit analysis, vendor reliability and many more “lities”.
- Decision making analysis is done using the Multiple Criteria Decision Making (MCDM) techniques [7] such as Weighted Score Method (WSM). It aids component evaluation.

2.3.2 Drawbacks of CRE

- In case of large no. of COTS alternatives and evaluation criteria, decision making process can be very complex due to the refinement of NFR’s.

- Exit criteria from the evaluation of all NFR's are not cleared.
- Mismatch Handling process of unsatisfied goal is absent**.**
- All drawbacks of WSM limit its use.

2.4 Off The Shelf Option (OTSO)

2.4.1 Description of OTSO

It was the first COTS selection technique proposed in 1995, which presents a method (as in figure 2.5) for evaluating and selecting Off-the-Shelf software components to be reused in software development. Following describes the main motivation and principles of the method and provides a detailed description of it.

- It provides a well defined, systematic and detailed evaluation criteria definition ([23], [24], [25]) based on functional requirements, product quality characteristics, strategic concerns, and architecture compatibility.
- It provides a method for estimating the relative effort and cost-benefits analysis of different alternatives [24].
- Weighted Score Method (WSM) [7] is there for better decision making for the evaluation of potential alternatives.

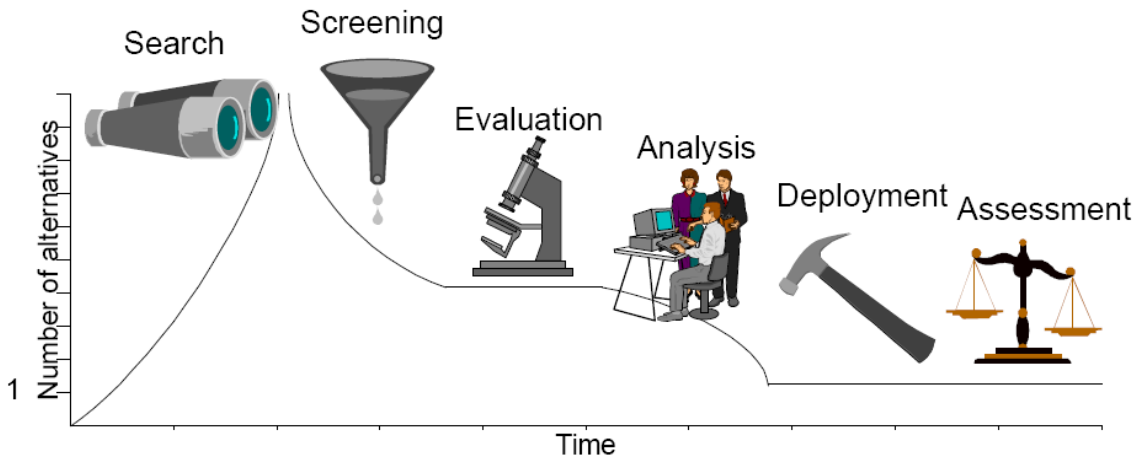

Figure 2.5: The Main Activities of the OTSO process [24]

- It reduces the time for COTS evaluation as it runs concurrently the evaluation criteria for the searched components.

- A method for comparing the “non-financial” aspects of alternatives, including situations involving multiple criteria [24].

2.4.2 Drawbacks of OTSO

- It directly starts with the evaluation criteria definition and fixed requirements as initially acquired.
- It doesn’t identify mismatches that arise between the requirements and the COTS features and resolution action for them.
- Drawbacks of WSM limit its use.

2.5 Mismatch-Handling aware COTS Selection (MiHOS)

2.5.1 Description of MiHOS

By A. Mohammed et al. ([26], [27], [28], [29]), MiHOS, (Mismatch Handling aware COTS Selection) a COTS selection approach distinguished from other approaches by its capability to address COTS mismatches under limited resource constraints. The main activities involved are pointed out here and depicted in figure 2.6:

- As name suggest, in this approach Mismatch handling is focussed in which mismatches and their resolution actions are analyzed; a selected set of mismatches is then resolved.

- It addresses COTS mismatches (not architectural mismatches) during two main phases of the selection process:

    - During COTS evaluation: It qualitatively handles mismatches and resolution will be just by refining the requirements and rest mismatches are postponed to next phase.

    - After COTS evaluation: mismatches are quantitatively addressed using matching level metric defined for each technical goal.

- COTS are selected on the basis of the maximum anticipated fitness value depending over the cost & effort required for resolving that mismatch.

- Weighted Score Method (WSM) [7] is there for better decision making for the evaluation of potential alternatives.

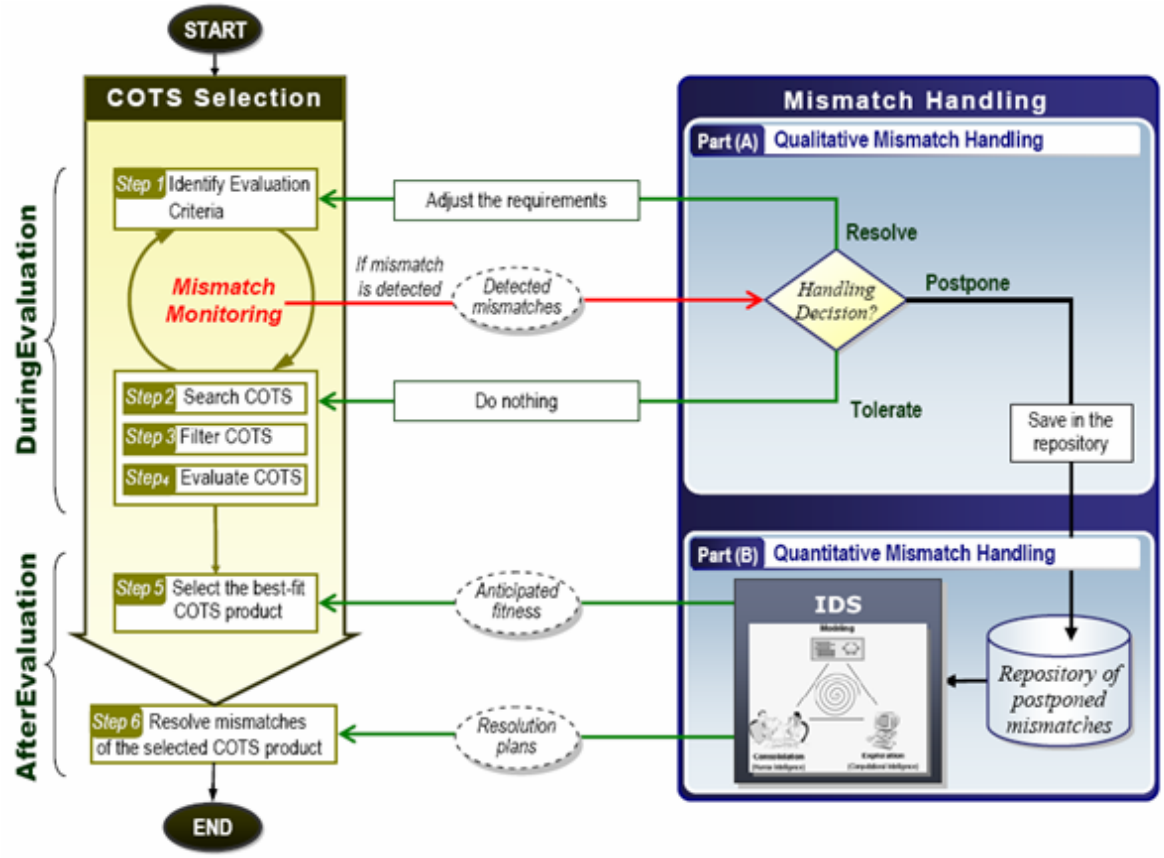

Figure 2.6: Two Phase Mismatch Handling in MiHOS [26]

- Interactive Decision Support like Sensitivity Analysis ([27], [28]) of the COTS is also performed for the metric like anticipated fitness and DIFF metric [28].
- MiHOS-PT tool is used to help in applying IDS-SA.
- It also presents a taxonomy of COTS mismatches and Matching Level (ML) metric [26] which is described below:

$$ML = \begin{cases} 0 & \text{iff there is a ZERO MATCH} \\ \in (0, 1) & \text{iff there is a PARTIAL MATCH} \\ 1 & \text{iff there is a MATCH} \\ \text{Undefined} & \text{for OVERMATCH \& SURPLUS} \end{cases}$$

2.5.2 Drawbacks of MiHOS

- Drawbacks of WSM (as discussed in section 2.6.1) are incorporated as the estimation of the weights to goals in Goal Graph is vague and subject to stakeholders.
- Assumption of the single resolution action for the identified mismatches.
- Non-functional requirements are ignored

2.6 Decision Making Techniques

When selecting a suitable COTS product, each COTS alternative should be ranked on how well it fits customers' requirements. Decision making techniques have been used in existing COTS selection methods for this purpose. The two most commonly used approaches are the Weighted Score Method and the Analytic Hierarchy Process.

2.6.1 Weighted Score Method

The WSM [7] method calculates the overall satisfaction for each product against the set evaluation criteria using the formula:

Total $Score_i = \sum (weight_j * Score_{ij})$ for i = 1 to n

where $weight_j$ is the weight of the $_j$th criterion, and $score_{ij}$ is the satisfaction score of the $_i$th alternative in terms of the $_j$th criterion.

The weights are assigned by stakeholders, and the satisfaction score represents the compliance of the product with a specific criterion [7]. An example is shown in Table 2.1, where the weights are represented on a 9-point scale:

COTS1 is selected as its total score, calculated using the formula given above, and has the maximum value. WSM technique is easy to apply and less time consuming.

Table 2.1: Example of Weighted Score Method

Criteria	Weight	COTS 1	COTS 2	COTS 3

Cost	5	1	0.5	0.5
Reliability	8	1	0.3	0.6
Performance	9	0.7	0.8	0.4
Usability	5	0.3	1	0.6
Security	9	0.8	0.6	0.7
Total Score		28.0	22.5	20.2

Drawbacks of WSM

- The results are represented by real numbers, which might be misinterpreted as difference between the candidates is very close.
- Estimation of the weights is difficult as it depends on the stakeholder's decision and also the number of criteria.
- Consolidation of the evaluation results into a single score is sometimes misleading because a high score in one attribute will hide a poor performance in another.

2.6.2 Analytic Hierarchy Process

The AHP method [4] is based on arranging the evaluation criteria in a hierarchy, descending from the overall goal, the criteria, the sub-criteria, and finally the COTS alternatives. The relative importance of the criteria at each level of the hierarchy is assessed by comparing them in pairs. Saaty et al. [13] introduced the following 9-point intensity scale which can be used in this comparison:

- 1 indicates that criteria C1 and C2 are of equal importance
- 9 indicates that criterion C1 is extremely more important than C2
- Any value between 1 and 9 represents different levels of relative importance, while reciprocals means C2 is more important than C1.

The results of the comparison are then converted into normalized rankings using an eigenvalue [13] technique on the comparison matrix. The normalized rankings represent the weights of the compared criteria. Similarly, the total weighted scores of

different products can be estimated by comparing these products in pairs with respect to each criterion.

Table 2.2 shows an example of applying AHP to weigh the four criteria at one level of the hierarchy. A pair-wise comparison is performed among the criteria, and the results are represented using the 9-point scale described above. For instance, criterion C1 in Table 2.2 is extremely more important than C2, while C4 is extremely more important than C1. The resultant normalized ranking (i.e. weight) of each criterion is shown in the right part of Table 2.2.

Table 2.2: Example of Analytic Hierarchy Process

Criteria	C1	C2	C3	C4	Normalised Value
C1	1	9	3	1/9	.208
C2	1/9	1	5	1/9	.074
C3	1/3	1/5	1	1/7	.043
C4	9	9	7	1	.675

Drawbacks of AHP

- Comparison involves stakeholder's decision which might be misleading.
- AHP involves many pair-wise comparisons, which would require a large amount of effort and time for a large number of criteria.

2.6.3 Feature Analysis using GQM paradigm

A Griman et al. [31], the Features Analysis makes two contributions; first it pinpoints some of the aspects that must be present in a COTS product and, second, it identifies the characteristics of the COTS evaluated. These strengths are determined by their dynamic behaviour in response to the features proposed. In order to identify the features of the evaluation, the steps indicated by Kitchenham [8] were taken as a reference in order to draw up lists of features. These are: identification of groups of users with different types of requirements, review of the initial list of requirements

that represent different groups that have already been identified, and lastly, identification of a suitable metric for each of them. This last step is based on the paradigm Goal Question Metrics (GQM) proposed by Basili [9]. Figure 2.7 depicts the hierarchy of goals, questions and their metrics.

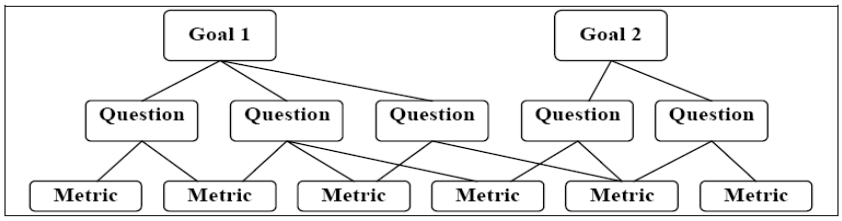

Figure 2.7: GQM paradigm by Basili [9]

A specific feature can be broken down into sub-features, which can be broken down further into conceptually simpler elements. These elements can be named attributes. Attributes are simply measurable physical or abstract quality of an entity [8]. A Gariman et al. [31] has used this technique for analysing and comparison of different Architecture Evaluation Methods like Design and Use of Software Architecture (DUSA), Architecture Trade off Analysis Method (ATAM), and Architectural Evaluation Methods (AEM).

2.7 Mismatch Handling

Unlike traditional development, in Reuse Based Development, simultaneous tradeoffs are performed among the COTS features, User requirements, System architecture. These trade-offs are performed in order to detect the mismatches between the COTS features & user requirements or architectural differences between the desired system and of the COTS. According to A. Mohammed et al. [26], two types of mismatches are encountered during component selection: - (I) *Architectural Mismatches*, which means difference in programming language, database types, calling procedures of different functions and (ii) *COTS Mismatches*, which means COTS features are not compatible with the stakeholder's requirements.

Yakimovich et al. [32] classified mismatches into two types: (i) wrong functionality, which means different functionality is incorporated in order to implement the required feature and (ii) missing functionality, which means the absence of a required feature.

Alves et al. [33] reformulated this work and investigated different types of issues caused by COTS mismatches. The 3 step framework by Alves (as given in figure 2.8) classifies COTS mismatches into three types: (i) D*iffer:* it means that there is a partial match between a COTS feature and a stakeholder's requirement. For example, a requirement states that "the COTS product shall mail the document as attachment and export as text or HTML file format," but a COTS product does not support the mail as attachment. (ii) *Extend:* It means that there are extra features present in the COTS when compared to the requirements. This excess might have helpful, neutral, or hurtful effects on the system. For example, a COTS product allows one user to open the files from remote location using FTP client but as not required. This extra feature might have a hurtful or helpful effect depending on the situation and (iii) *Fail:* This type occurs when there is a zero match between the COTS product and requirements. This type is similar to the missing functionality mismatch in Yakimovich's work [32].

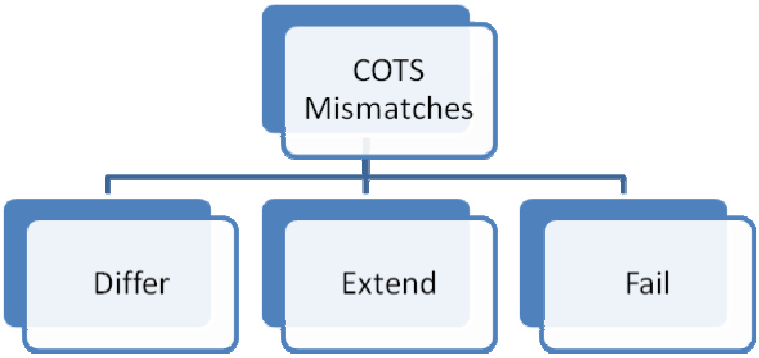

Figure 2.8: Mismatch taxonomy proposed by Alves

MiHOS, a COTS selection approach proposed by A. Mohammed et al [26] has defined relationship between the COTS features and the technical goals as MATCH and MISMATCH. The further classification is given in figure 2.9. He also proposed Matching Level metric as defined in subsection 2.5.1 above.

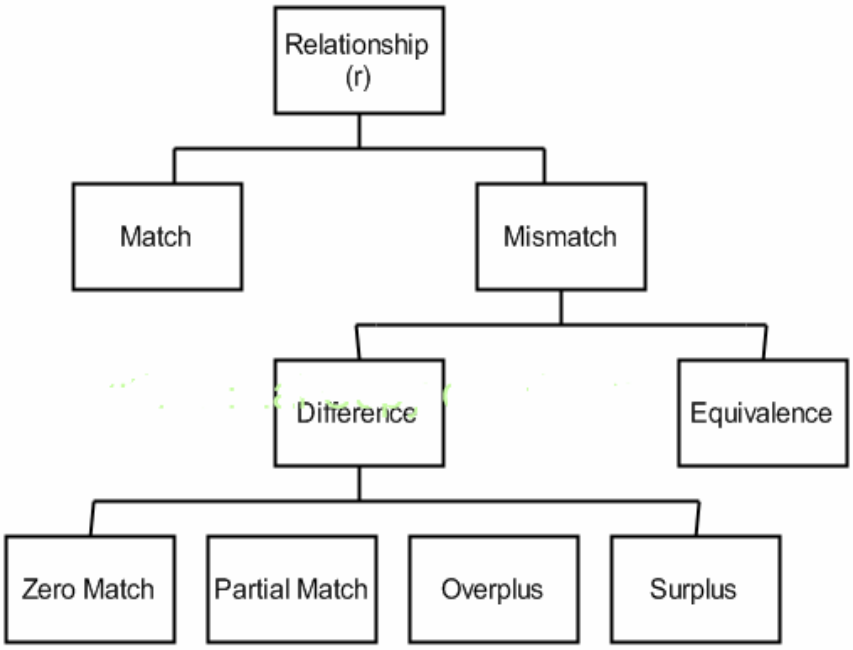

Figure 2.9: Mismatch taxonomy proposed by MiHOS [26]

2.8 Summary

This chapter has presented state of the art in COTS selection. It began with the description of the 5 COTS selection process and their drawbacks also. Then it provided an overview of the two mostly used decision making techniques viz AHP & WSM. Then the overview of the GQM based Feature Analysis technique. The next chapter describes the proposed COTS selection process using GQM based Feature Analysis technique which incorporates mismatch handling also.

Chapter 3

Problem Statement

In previous chapter we explored five COTS selection processes for the development with reuse. We have also seen advantages and disadvantages of all the techniques. But no technique is fully efficient. Because some of the process like CARE, PORE, supports the partial mismatch handling but the whole process become complex for large no. of COTS alternatives whereas MiHOS supports mismatch handling in two phases but it also incorporates the drawbacks of WSM. COTS are represented in functional, non-functional and architectural attributes in CRE but the whole process is not well defined as it doesn't talk about the exit criteria from NFR framework. OTSO presented a selection process with well defined evaluation criteria but doesn't support the changing requirements. All the above processes are compared against the COTS selection criteria (C1-C7) defined in sub section 1.6.1 with the help of the table 3.1.

Table 3.1: Comparative study of COTS selection processes.

Criteria → CSP ↓	C1	C2	C3	C4	C5	C6	C7
OTSO	C	C	B	C	B	A	C
PORE	A	B	C	B	C	C	B
CARE	A	B	C	B	B	C	B

CRE	A	B	A	A	B	B	C
MiHOS	A	B	B	C	B	B	A

In [30] A. Mohammed et al. has surveyed and done the comparative analysis of 18 most of the significant approaches (from 1995-2006) that contributed to the improvement of current COTS selection practices and also highlighted some open research issues relevant to the selection process, and concluded with a discussion of possible future directions to address these issues.

In the previous chapter, Feature analysis technique using GQM paradigm is also explored in which refinement of goals into features and sub-features is done and operational questions with their quantitative metrics are formulated. These metrics would help in evaluating the operational characteristics of the COTS components.

Our aim in this thesis work is to propose a COTS selection process with the help of the GQM based Feature Analysis technique to do COTS evaluation. We also aim to identify maximum no. of mismatches between the COTS capabilities and the gathered requirements so as to decrease the time and effort spent over the COTS integration.

Chapter 4

Proposed COTS Selection Process

As we have discussed in the chapter 3, many COTS selection techniques are already proposed but they have some limitations also. Most of the techniques make use of WSM and AHP as a decision making technique for the evaluation of different COTS products. But these techniques come with drawbacks also. AHP involves many pair-wise comparisons, which would require a large amount of effort and time for large no. of components. In WSM, estimation of weights to features or sub-features is very vague in nature. Techniques like CARE, PORE handles mismatches partially. In order to overcome these aspects, the proposed model focuses on GQM based Feature Analysis of COTS products in order to have the detailed evaluation of most promising candidates.

4.1 The Systematic Process

The systematic process of the proposed COTS selection process has been divided into six phases (as in figure 4.1):-

- Requirement Acquisition
- Searching
- Filtering
- Detailed Evaluation with Dynamic Analysis

- Mismatch Handling
- Ranking the Components

4.1.1 Requirement Acquisition

Requirement forms the basis of any effective COTS selection techniques as they provide the acceptance criteria of any COTS. Requirements are elicited from various stakeholders on the basis of the various requirement elicitation techniques like interviewing and brain storming session. This phase ends with the requirement specification document.

4.1.2 Searching

In this phase, potential reusable candidates are identified. In initial phase, keyword based search is performed over many COTS sources. These sources can be organisation specific in-house libraries or over the internet (For Eg: Google.com, yahoo.com, infoseek.com) as they contains the updated information about the commercial, shareware and freeware software as well. Search results in the name of the potential components, their www address, vendor history, product licensing features and characteristics.

The search should be initiated as soon as main requirements or the basic features are known from the stakeholder's side. Search and requirement acquisition goes parallel and thus entertains the requirements discovered at later stage.

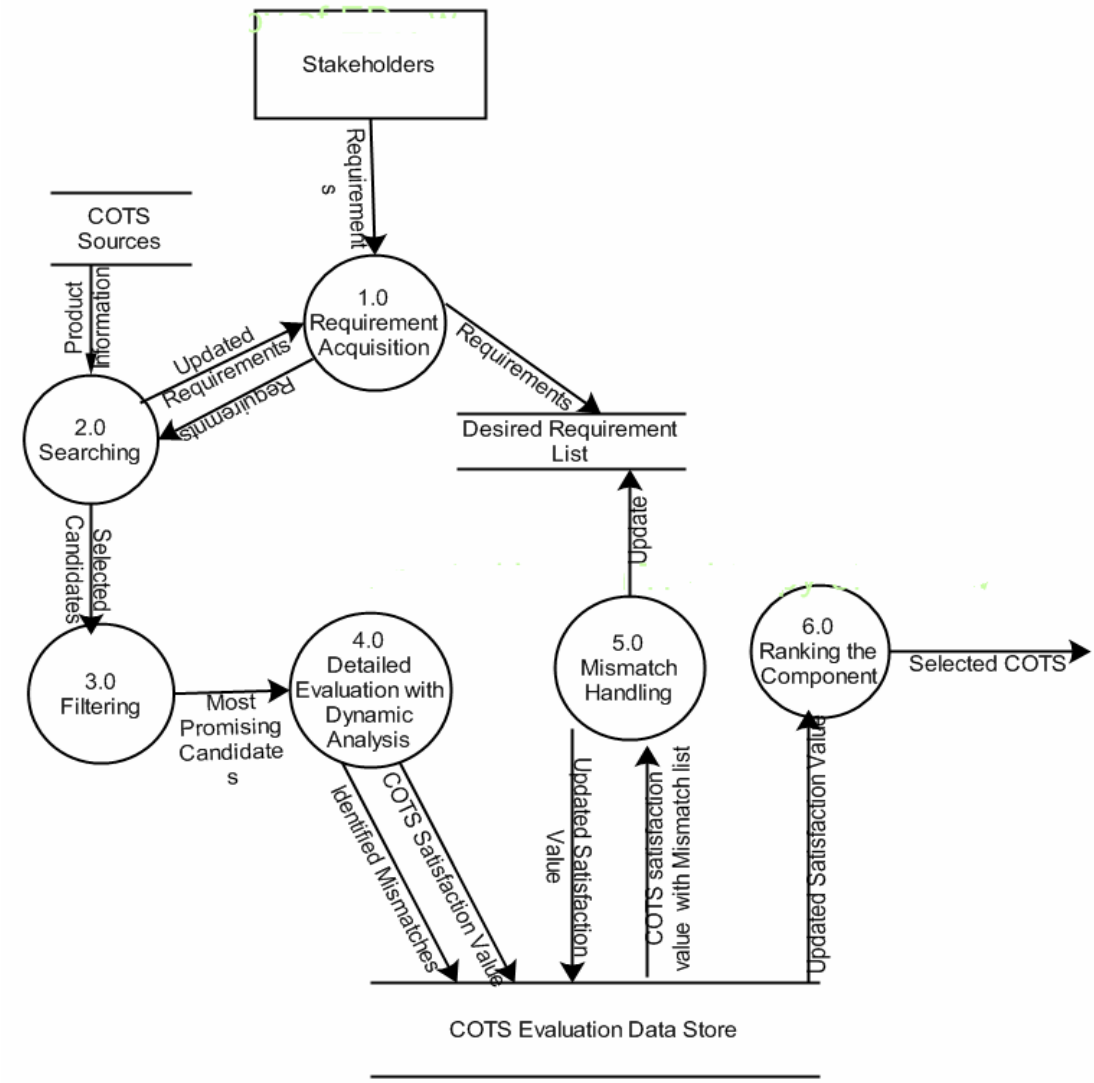

Figure 4.1: The Overview of the Proposed COTS Selection Process

4.1.3 Filtering

In this phase, identified potential candidates are examined with the aim of getting most promising candidates. Basically, list of searched candidates is filtered out so as to reduce the time and effort required for the detailed evaluation of each and every COTS. Filtering is performed using the criteria based on the requirements like Operating system support, programming language difference, and cost factor. Moreover, the preparation of requirement checklist using the COTS information gathered in the search phase is also helpful for the filtration. COTS with maximum availing functionality are filtered out and act as most promising candidates.

4.1.4 Detailed Evaluation with Dynamic Analysis

The aim of this phase is to evaluate the candidates come out as a result of filtration, on the basis of their behaviour captured during their dynamic analysis using pilots, trial version, hands on demonstrations or execution. The evaluation results are then documented. Dynamic analysis works well with COTS components as they are black box in nature and available with the .exe files, design specification without source code and trial demonstrations. So, the technique applies straightforwardly to COTS components, which are particularly difficult to address with traditional testing and analysis techniques.

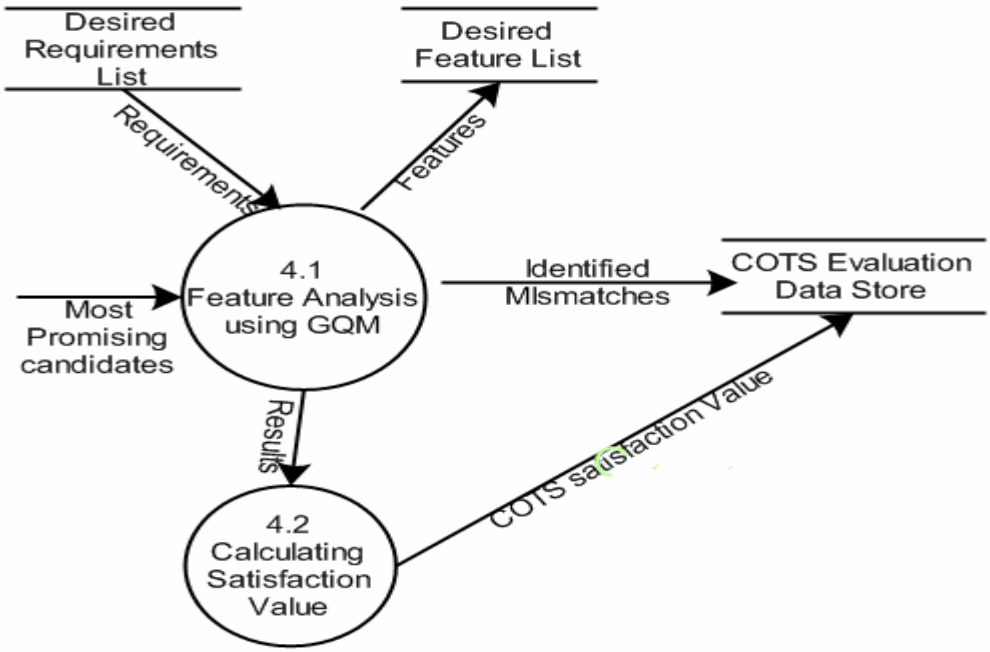

Figure 4.2: Detailed Evaluation of the Promising Candidates

Detailed evaluation runs into 2 following sub activities as explained in figure 4.2 using data flow diagram:

- Feature Analysis using GQM
- Calculating Satisfaction Value

Feature Analysis using GQM Paradigm

It overcomes the drawbacks incorporated by AHP or WSM for decision making in COTS selection. It makes two contributions; first it pinpoints some of the aspects that must be present in a COTS product and, second, it identifies the characteristics of the COTS evaluated. These strengths are determined by their dynamic behaviour captured during the Dynamic analysis in response to the desired features based on the GQM paradigm.

GQM based feature analysis template, for example, richness of built-in tools, is given in table 4.1. Once the features, sub features and their operational questions are identified, metric should be defined for each operational question in order for the evaluation of the COTS products and able to get the satisfaction value of the different COTS. Two types of metrics defined for the measurement of operational questions are: *Ordinal Metric* and *Binary Metric*. As name suggest, ordinal metric is defined using ordinal scale and binary metric is defined using nominal scale. These metrics are clearly explained in the appendix A.2.

Calculating Satisfaction Value

To calculate Satisfaction Value of each promising COTS, normalization of the ordinal values assigned to sub features of promising COTS in the ordinal scale should be done. For the normalization process, satisfaction function is defined which calculates the satisfaction values for all the features of each COTS candidate.

Table 4.1 GQM Based Template for "Richness of Built-In Tools"

Feature	Functional	
Sub-Feature	Richness of Built-In tools	
Description	The availability and variety of extra tools built in the system	
Operational Questions		***Used Metric***
1. Does it offer a feature in order to have the spell check of the document?		BIN
2. Does it offer a tool to have the statistics of the words present in the document?		ORDN
3. Is it able to mail the document as attachment/ text or export it?		ORDN
4. Is the system having the Multi-view feature in order to have the comparison?		ORDN
5. Does the system support UNICODE Encoding scheme?		BIN
6. Is the system having the session management feature?		BIN

Satisfaction Function

Satf: Set of Ordinal Values -> [0, 1] as

Satf (x_i) = a*x_i + b where the given initial conditions are:

Satf (0) = 0

Satf (3) = 1

$$\begin{cases} 1 \text{ if the impact is neutral,} \end{cases}$$

Satf (4) = or helpful

0 if the impact is hurtful

These initial conditions compute a=1, b=0 and Satf = $x_i/3$ where x_i is the ordinal value assigned as per the scale defined in appendix A.2.

Satf (COTS) = $\sum$ Satf (x_i)

This COTS evaluation data is then properly documented into COTS evaluation data store.

4.1.5 Mismatch Handling

Mismatch handling process is subdivided into 4 activities (as explained in figure 4.3):

- Classify Mismatches
- Suggest Mitigation Action
- Refine Feature
- Set Mismatch Value: 0

MV (x_i) = 1-Satf (xi) for each features of COTS defined in the template, where MV (x_i) stands for Mismatch Value.

Mismatches identified in the previous phase are analyzed and the decision about their handling is taken. On the basis of the taken decision, mismatches are classified into 3 categories:-

Resolvable Mismatches – In this category mismatches are resolved by customising the COTS functionality using any mitigation action as discussed in sub-section 4.2.2. For example, “Is the COTS able to mail the document as attachment or export it as HTML/ Text”. Like the COTS7 viz Notepad++ is able to send the document via outlook but not able to export it as HTML/ Text file. By adding plug-in NppExport to the Notepad++, this mismatch is resolved.

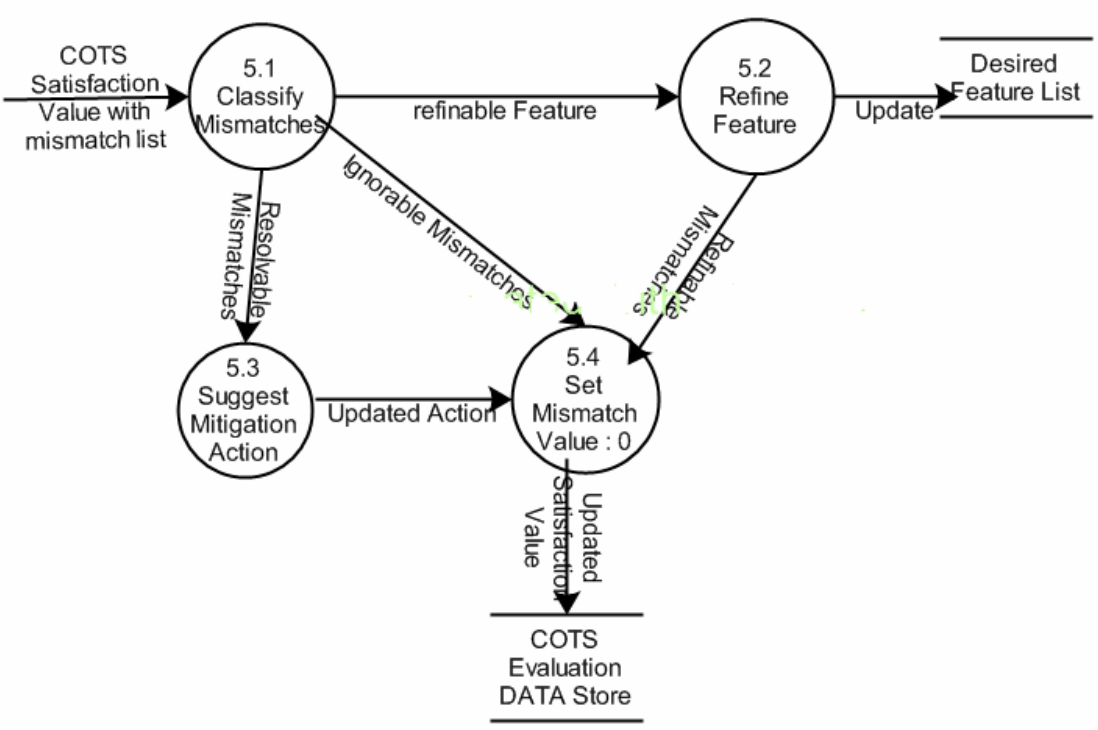

Figure 4.3: Data Flow Diagram for Mismatch Handling Process

Refinable Feature –This case arises when the corresponding feature is absent but the said feature is covered by another COTS capability. For example, in Programmer's Notepad the desired feature "Does the system provide free support by expert" can be refined as it provides help through discussion forums.

Ignorable Mismatches- On the basis of the Mismatch value and their impact over the application, mismatches are ignored. It is subjective to the stakeholders' decision.

By suggesting the mitigation action for the respective resolvable mismatches or for the ignorable mismatches, the mismatch value is changed to zero. Corresponding mitigation action and new mismatch values (MV_{new}) are updated for the respective COTS in the COTS Evaluation Data Store.

4.1.6 Ranking the Components

Calculate new satisfaction value for each and every most promising candidate. Ranking of COTS on the basis of the new satisfaction value after handling the mismatches is then done.

Satf(COTS) = $\sum$ (1-MV_{new} (x_i))

% of Resolution of Mismatches = ($\sum$ MV_{new} (x_i) / $\sum$ MV (x_i))* 100

4.2 Mismatch Mitigation Action

COTS mismatches can be mitigated by taking following 2 mitigation action:-

4.2.1 Refining Requirements

This strategy suggests modification in the requirements, and thus making a compromise that reduces the mismatch. This action is taken when the desired feature is not present in the COTS but another COTS attributes satisfies the requirements. However, the current COTS selection methods only address the COTS mismatch problem during the requirements definition phase, but these approaches do not talk about the effect of mismatches on COTS selection, nor do they talk of mitigation action for the remaining violations. But in this thesis work this has been considered.

4.2.2 Resolvable Mismatches

This strategy is used to resolve mismatches that are related to missing COTS functionality. According to Vigder et al. [5] this can be realized by several alternatives such as:

- Acquiring additional plug-ins that add functionality to the COTS product,
- Writing custom code using a scripting language supported by the COTS, for example JavaScript, Visual Basic, and Perl.
- API – Application Programming Interface that is an interface implemented by a software program which enables it to interact with other COTS software.
- Parameter based Tailoring
- GUI based Tailoring
- Modifying Source Code

4.3 Summary

In this chapter, we have proposed a six step- formal process for the COTS selection so as to handle the mismatches raised between COTS capabilities and the desired features in early phase of the COTS based development. At the end of this chapter,

various mismatch mitigation actions are also discussed. The next chapter describes its illustrations and feasibility.

Chapter 5
Experimental Results

The specific goal of the case study (Source Code Editor) is to validate the feasibility of the proposed model for the COTS selection. These are the editors which facilitate the editing of source code written in different programming languages and incorporate basic editing features also. They help programmers to write effective code in short duration as it offers many features like syntax highlighting, code folding, running of scripts etc. Study shows how the process is used to select the best Source Code Editor having the maximum satisfaction value with the desired set of requirements among the other alternatives.

5.1 Application of Proposed COTS Selection Process

5.1.1 Requirement Acquisition

It results in various functional and non-functional requirements (as in figure 5.1) formulated in requirement specification document as the output. Requirement Engineers and users are involved in this phase.

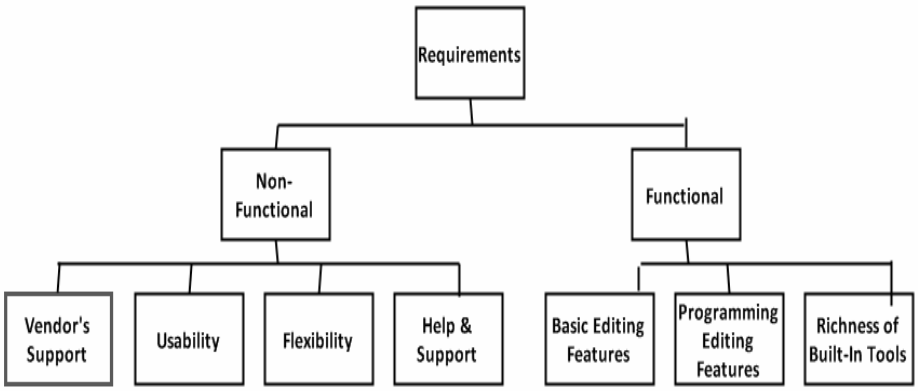

Figure 5.1: Hierarchy of the requirements

5.1.2 Searching

In order to identify possible COTS candidates based on the gathered requirements, we have used several information sources, such as online search engines like Google and Yahoo, as well as specialized websites that list and categorize source code editor products; e.g. www.cnet.com. This resulted in identifying an initial set of 15 COTS candidates. Information about their names, www address, vendor name and version is tabulated in the table 5.1 and the snapshots are given in the appendix C.2.

Table 5.1: List of Searched Candidates

COTS ID	Product Name	Version	Website	Vendor
COTS 1	Caderno	2.10.1	Freddy.com	Freddy Team
COTS 2	Crimson Editor	3.70	Crimsoneditor.com	Emerald Editor
COTS 3	Edit Plus	3.12	Editplus.com	Sangil Kim
COTS 4	Editpad lite	6.3.1	Editpadpro.com	Edit Group
COTS 5	Em Editor	9.01	Emeditor.com	Emurasoft
COTS 6	Notepad 2	4.0.23	Sourceforge.net	Florian Balmer
COTS 7	Notepad++	5.6.8	Sourceforge.net	Don Ho
COTS 8	Notepaddotnet	0.2	Dotnet-notepad.com	Moimael
COTS 9	NoteTab Standard	6.20	Notetab.com	Fookes Software
COTS 10	Powerpad	1.2.1	Softpedia.com	Natham Osman
COTS 11	Programer's Notepad	2.1.4	Pnotepad.org	Simone Stelle
COTS 12	PSPAD	4.5.4	Pspad.com	Jan Fiala
COTS 13	Ted Notepad	5.3.1	Jsimlo.sk/notepad/	Juraj Simlovic
COTS 14	Text Edit	1.4	Softpedia.com	Apple Computer
COTS 15	Total Edit	5.5	Codertools.com	Codertools

5.1.3 Filtering

The evaluators collected detailed information about these 15 candidates from their vendor's websites or other online sources which provide information about them.

Figure 5.2: Snapshot for Notepad ++

Then evaluators filtered out the most promising 5 candidates (snapshots are given in appendix C) out of 15 candidates based on the screening rules defined as the result of the brain storming session**.**

Table 5.2: Shortlist of the most Promising Candidates

COTS ID	Product Name	Version	Website	Vendor
COTS 7	Notepad++	5.6.8	Sourceforge.net	Don Ho
COTS 11	Programer's Notepad	2.1.4	Pnotepad.org	Simone Stelle
COTS 12	PSPAD	4.5.4	Pspad.com	Jan Fiala
COTS 2	Crimson Editor	3.70	Crimsoneditor.com	Emerald Editor
COTS 9	NoteTab Standard	6.20	Notetab.com	Fookes Software

5.1.4 Detailed Evaluation with Dynamic Analysis

The objective of this phase is to evaluate the list of shortlisted candidates as an input to this phase from the previous phase. Evaluation of all the selected 5 candidates on the basis of the 7 features and 36 operational questions or sub-features proposed based on GQM paradigm (as discussed in detail in appendix B) is done by the evaluators by dynamically executing all the 5 candidates or using the pilot programs or demonstrations and then scoring is done in parallel using the metrics defined in appendix A.2. For example, the operational question "Is the system provides the Syntax Folding feature".

Metric Type: BIN

Value: 0 means the feature is absent in COTS 2.

Type: FALSE

Normalised Satisfaction Value: 0.00 indicates no satisfaction between the COTS 2 (Crimson Editor) capability and the required operational question.

Features	GQM Based Operational Questions	Metric Type	COTS 2 (Crimson Editor)			COTS 7 (Notepad++)			COTS 11 (PN2)			COTS 9 (Notetab-Std)			COTS 12 (PSPAD)		
			Value	Satisfaction Value	Type	Value	Satisfaction Value	Type	Value	Satisfaction Value	Type	Value	Satisfaction Value	Type	Value	Satisfaction Value	Type
	Does the system have the Zoom-in/ out facility?	BIN	0	0.00	FALSE	1	1.00	T									FALSE
Programming Editing Features	Does it offer a technique to highlight the syntax for various languages like C, C++, JAVA, PHP, Perl, Python?	BIN	1	1.00	TRUE												
	Is the system provide the Syntax Folding feature?	BIN	0														E
	Does it offer Code comment/ uncomment feature?	BIN	0	0.00	FALSE	1	1.00										TRUE
	Does the system provide compilation & execution of the scripts?	BIN	1	1.00	TRUE	0	0.00	FALSE	0	0.00				FALSE	1	1.00	TRUE
	Does the system provide Bracket Matching?	BIN	0	0.00	FALSE	0	0.00	FALSE	0	0.00	FALSE	0	0.00	FALSE	1	1.00	TRUE
	Does it offer a feature in order to have the spell check of the document?	BIN	0	0.00	FALSE	0	0.00	FALSE	0	0.00	FALSE	0	0.00	FALSE	1	1.00	TRUE

Evaluation of COTS2 against "Syntax Folding Feature"

Metric Type: BIN

Value: 0

Satisfaction Value: 0.00

Figure 5.3: Detailed Evaluation Results using GQM based Feature Analysis

In figure 5.3, it addresses the question “How the mismatches are detected” and their metric type and normalised satisfaction value. For the complete results of each operational question refer to appendix C.

The corresponding satisfaction values for each COTS are tabulated in table 5.3 below:-

Table 5.3: Summary of Evaluation Results before Mismatch Handling

COTS ID	Satf (COTS) (Before Mismatch Handling)	Count of Mismatches	% of Mismatches to the no. of Operational Questions (36)	Rank (Before Mismatch Handling)
COTS 2	20.67	18	50 %	4
COTS 7	28.67	10	27.7%	2
COTS 11	27.33	12	33.3 %	3
COTS 9	14.33	24	66.6 %	5
COTS 12	30.67	7	19.4 %	1

5.1.5 Mismatch Handling

As per the results shown in table 5.3, COTS 2 and COTS 9 are not of much concern as they involve maximum no. of mismatches. Thus we are taking COTS 7, COTS 11, COTS 12 under the consideration for this phase. As per the process, we have classified the identified mismatches into the resolvable, ignorable mismatches and the refinable features. Firstly we have calculated the

Mismatch Value MV (x_i) = 1- Satf (x_i) for all the promising candidates.

Mitigation Action is chosen in the stakeholder’s (project manager & evaluator) discussion session for the respective mismatches. New Mismatch value is assigned to the resolved mismatches or ignored mismatches as 0. Finally compute the new Satisfaction value of each feature and then of respective COTS as follows:

$Satf_{new}$ (COTS) = $\sum$ (1- MV (x_i)) + (Total no. of Features - Count of Mismatches) * 1

For example, in Notepad++, mismatches corresponding to mismatch value 1 (spell check, export, compare, statistics as in the appendix (C.3.1) are resolved by adding the plug-ins. Figure 5.4 shows how the mismatches can be resolved using plug-ins.

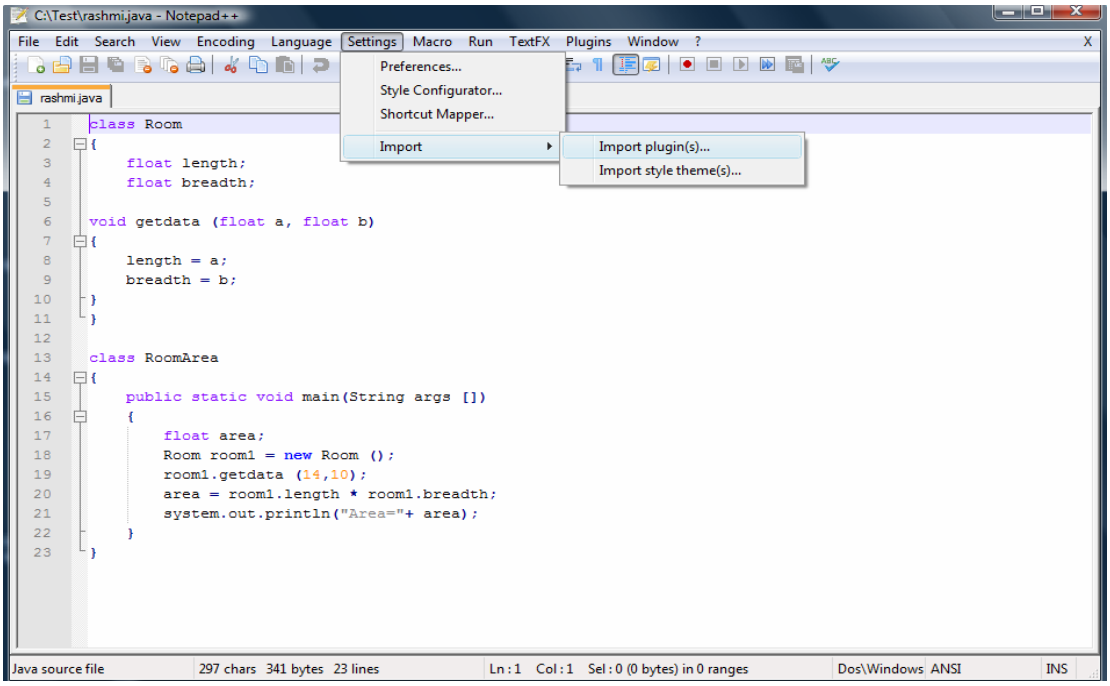

Figure 5.4: Importing Plug-ins in Notepad++ Part (1/2)

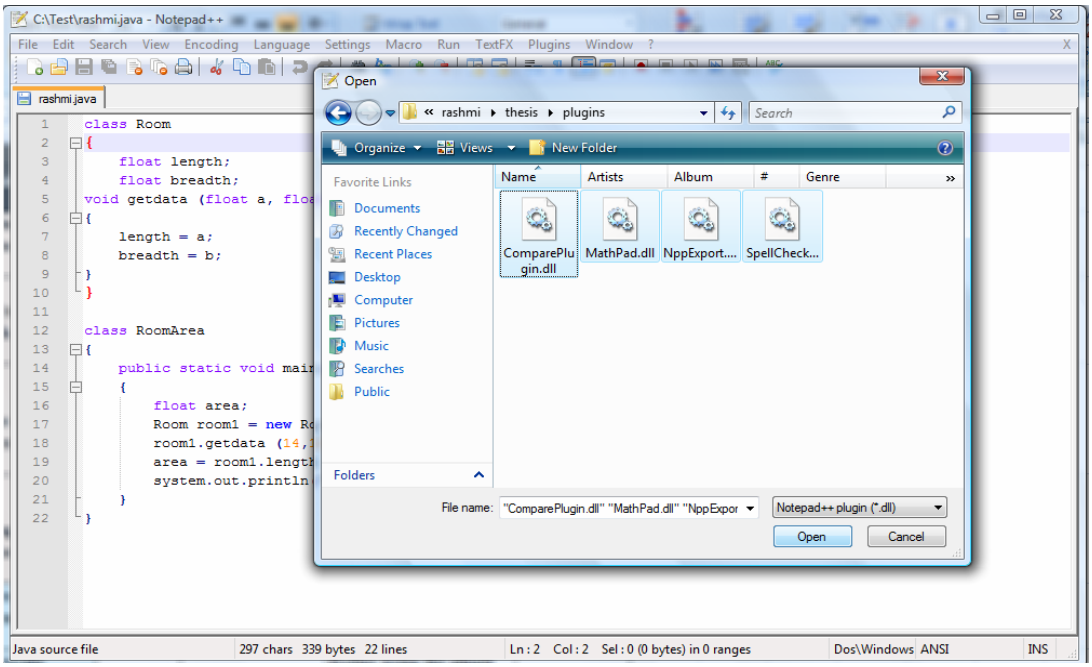

Figure 5.4: ComparePlugin.dll, Mathpad.dll, Nppexport.dll, and SpellCheck.dll Part (2/2)

Table 5.4: Results after Mismatch Handling

COTS ID	Total Mismatch Value (MV)	Total New Mismatch Value (MV_{new})	% of Mismatch Resolution	New Satisfaction Value
COTS 7	7.33	1	86.33	35.0
COTS 11	8	3.99	50.1	32.0
COTS 12	5.33	2.67	49.99	33.33

5.1.6 Ranking the Components

As per the results in table 5.3, Notepad++ was at rank 2 and the no. of mismatches were greater than that of PSPAD editor. But after mismatch handling as in table 5.4, satisfaction value has increased for Notepad++ (35) because of the capability of resolving mismatches as compared to that of PSPAD (33.33).

So, Notepad++ is selected for our application of Source Code editor.

5.2 Summary

Before Mismatch handling, as in table 5.3, PSPAD Editor got rank 1 with the count of mismatches 8 whereas Notepad++ came at 2 with 10 mismatches. But after applying resolution actions like plug-in for resolving these mismatches, notepad++ moved to the position 1 due to the presence of the tool “Add Plug-in”.

Chapter 6

Conclusions and Future Scope

The aim of this thesis is to propose a COTS selection process which helps in detection and resolution of the mismatches between the COTS capabilities and stakeholder's requirement during the COTS selection so as to decrease the time and effort spent over the COTS integration. We have proposed a six-step COTS selection process in which the evaluation of COTS is GQM paradigm based. Mismatch mitigation actions are suggested for the identified mismatches belonging to resolvable category. Refinement of the requirement is carried in case if the requirement is satisfied by another feature of the COTS.

6.1 Conclusions

As the evaluation of COTS in the proposed process is carried out on the basis of GQM based Feature Analysis, which resolves the drawbacks of WSM and AHP that we have discussed in chapter 2, the efficiency of detecting the right mismatches is better in the proposed process. Major conclusions derived from the work undertaken are:

- GQM based Feature Analysis helps in scoring the match between the COTS capability and required features to the granular level.
- Mismatches are not deferred till the integration phase as it detects most of them and suggest mitigation action for each mismatch during the COTS evaluation only.
- It also covers functional and non-functional aspects of the system under development compared to MiHOS where they are ignored.

6.2 Future Scope

- The initial keyword based search can be replaced by any other COTS retrieval technique so as to retrieve more relevant components.

- As the process focuses on the single COTS software based system, so, it needs to address multiple COTS selection. For that, issues like interoperability between COTS and architectural mismatches need to explore.
- Applying the process to different case studies of various domains is necessary in order to evaluate it more deeply.
- The current work can be supported with the help of a tool to make the selection more effective.

References

[1] I. Crnkovik, S. Larsen and M. Chaudron, “Component-based Development Process and Component Lifecycle”, 2001.

[2] H. Mili, A. Mili, S. Yacoub, and E. Addy, “Reuse Based Software Engineering”, Wiley-Interscience Publication, USA 2002.

[3] M. R Vigder, J. Dean, “An Architectural Approach to Building Systems from COT Software Components”, National Research Council (NRC), 40221.

[4] M. Torchiano and M. Morisio, "Overlooked Aspects of COTS-Based Development," IEEE Software, vol. 21 (2), 2004 pp. 88-93.

[5] M. R. Vigder, W. M. Gentleman, and J. Dean, "COTS Software Integration: State of the art," National Research Council Canada (NRC), 39198, Jan 1996.

[6] T. Wanyama and B. Far, “An Empirical Study to Compare three Methods for selecting COTS Software Components”, IJCIR, Vol 2 no.1, pp 34-46, June 2008.

[7] T. Wanyama and B. Far, “Towards Providing Decision Support for COTS Selection”

[8] B. Kitchenham and L. Jones, 1997, “Evaluating Software Engineering Methods and Tools: Part 5, The Influence of Human Factors”, Software

[9] V. Basili, “Software modeling and measurement: the goal/question/ metric paradigm” In Technical Report CS-TR-2956, University of Maryland, 1992.

[10] C. Ncube and N. A Maiden, “Procuring Software Systems: Current Problems and Solutions”, Proceedings REFSQ97 workshop, CaiSE97, Barcelona, Spain, June 16-17 Engineering Notes 22(1).

[11] G. Fox, S. Marcom and K. Lantner, “A Software Development Process for COTS-based Information System Infrastructure”, Proceedings of the 5th International Symposium on Assessment of Software Tools and Technologies (SAST’97), pp133-143, 1997.

[12] C. Ncube and N. Maiden, “PORE-Procurement Oriented Requirement Engineering Method for the Component-Based Systems Engineering

Development Paradigm", International Workshop on Component Based Software Engineering, 1999.

[13] T. L. Saaty, "The Analytic Hierarchy Process", New York, McGraw-Hill 1990.

[14] L. Chung, K. Cooper and S. Courtney, "COTS-Aware Requirements Engineering: The CARE Process", In Proc. of the 2nd International Workshop on Requirements Engineering for COTS Components, Kyoto, Japan, September 7, 2004.

[15] L. Chung, K. Cooper and D. Huynh, "COTS- Aware Requirement Engineering Technique", IEEE International Symposium on Requirements Engineering, 2002.

[16] L. Chung and K. Cooper, "A COTS-Aware Requirements Engineering Process: a Goal- and Agent-oriented Approach", In Proc. of the International Council on Systems Engineering Symposium, Las Vegas, Nevada, CDROM index 3.5.4, July 28 – August 1, 2002, pp. 1-8.

[17] L. Chung and K. Cooper, "A Knowledge-based COTS Aware Requirements Engineering Approach", In Proc. Of the Software Engineering and Knowledge Engineering Conference, Ischia, Italy, July 15-19, 2002, pp.175-182.

[18] L. Chung and K. Cooper, "Defining Agents in a COTS Aware Requirements Engineering Approach", Proc., 7th Int. Australian Workshop on Requirements Eng., 2002.

[19] L. Chung, and K. Cooper, "Defining Goals in a COTS-Aware Requirements Engineering Approach", System Engineering journal, 7(1), 2004, pp. 61-83.

[20] J. Mylopoulos, L. Chung, and B. Nixon, "Representing and using non-functional requirements: a process-oriented approach", IEEE Transactions on Software Engineering, Volume: 18 Issue: 6, June 1992, pp. 483 –497

[21] C. Alves and J. Castro, "CRE: A Systematic Method for COTS Selection" In XV Brazilian Symposium on Software Engineering, Rio de Janeiro, Brazil, October 2001.

[22] C.Alves, J. Castro and M. Alencar, "Requirements Engineering for COTS Selection" In III Workshop de Engenharia de Requisitos, 2000.

[23] J. Kontio, "OTSO: A Systematic Process for Reusable Software Component Selection" In UMIACS-TR-95-63, 1995.

[24] J. Kontio, "A Case Study in Applying a Systematic Method for COTS Selection", In Proc. of ICSE-18, 1996, pp.201–209.

[25] J. Kontio, S. F. Chen, and K. Limperos, "A COTS Selection Method and Experiences of Its Use", In Twentieth Annual Software Engineering Workshop, 1995

[26] A. Mohamed, G. Ruhe, and A. Eberlein, "MiHOS: An Approach to Support Handling the Mismatches between System Requirements and COTS Products " Int. Requirements Engineering Journal, Accepted on Jan 2, 2007 (DOI: 10.1007/s00766-007-0041-5).

[27] A. Mohamed, G. Ruhe, and A. Eberlein, "Decision Support for Handling Mismatches between COTS Products and System Requirements", in the 6th IEEE International Conference on COTS-based Software Systems (ICCBSS'07),Banff, Canada, 2007, pp. 63-72

[28] A. Mohamed, G. Ruhe, and A. Eberlein, "Decision Support for Customization of the COTS Selection Process," ACM SIGSOFT Software Engineering Notes Vol 14 (4), in Proc. of ICSE'05 workshop on Models and Processes for the Evaluation of COTS Components (MPEC'05) 2005, pp. 1-4.

[29] A. Mohamed, G. Ruhe, and A. Eberlein, "Exploratory Analysis for the Mismatch Handling Process" in preparation for submission to Int. Requirements Engineering Journal

[30] A. Mohamed, G. Ruhe, and Armin Eberlein, "COTS Selection: Past, Present, and Future," in the 14th IEEE International Conference and Workshop on the Engineering of Computer Based Systems (ECBS'07), Tucson, Arizona, 2007

[31] A. Griman, M. Perez, L. Mendoza and F. Losavio, “Feature analysis for architectural evaluation methods” In The Journal of Systems and Software 79 (2006) 871–888

[32] D. Yakimovich, J. M. Bieman, and V. R. Basili, “Software architecture classification for estimating the cost of COTS integration”. Los Angeles, California, United States: IEEE Computer Society Press, 1999.

[33] C. Alves and A. Finkelstein, "Investigating Conflicts in COTS Decision-Making," International Journal of Software Engineering and Knowledge Engineering (IJSEKE), vol. 13 (5), 2003 pp. 473-493.

[34] N. Fenton and S. L. Pfleeger, Software Metrics: A Rigorous and Practical Approach, Secon ed. London, UK: International Thomson Computer Press, 1997.

List of Papers

[1] Rashmi Gupta, Shivani Goel, "Comparative Study of COTS Selection Techniques", published in the National Conference on Emerging Trends in Information Technology and Computing (ETIC10), GITM, Gurgaon, India (27^{th} March 2010).

[2] Rashmi Gupta, Shivani Goel, "GQM based Mismatch Handling aware COTS Selection", communicated in the Communication Magazine, IEEE Xplore-Software.

Appendix A

Types of Measurement in Proposed Model

A.1 Measurement Scales

In general, Fenton and Pfleeger [34] have defined five measurement scales:

- *Nominal,* in which values are assigned to different categories. The values of a nominal scale have no quantitative meaning; rather they are qualitative and unordered in nature. For example, “sex” might be measured on a nominal scale {“male”, “female”}.

- *Ordinal,* in which a set of ordered values is used to measure an entity. But, the difference between adjacent values is not fixed. For example, “technical support” might be measured on an ordinal scale {“good”, “fair”, “bad”}, where we know that “good” is better than “fair”, but the distance between them is unknown.

- *Interval*, which preserves the order as well as the differences. But multiplication and division are meaningless here. For example, Fahrenheit scale for temperature. Equal difference between 2 temperatures remains same but the division & multiplication is not allowed. Like, 30 degree temperature is not twice as much of 15 degree temperature.

- *Ratio,* which is similar to interval scales except they have true zero point; i.e. it preserves division & multiplication also. For example, a 40 year-old person is twice as old as a 20-year old one.

- *Absolute,* which simply counts the number of occurrences of an element in an entity. For example, number of LOC is measured by counting the number of lines in an application.

A.2 Metrics in Proposed Model

A.2.1 Binary Metric (BIN)

These are the basic measure of any COTS evaluation process. BIN metrics are used to indicate the presence or absence of a COTS feature. A metric x that is BIN-based is estimated on a nominal scale, x ∈ {TRUE, FALSE}.

$$\text{Value Given} = \begin{cases} 0 \text{ iff } x = \text{“FALSE”} \\ 1 \text{ iff } x = \text{“TRUE”} \end{cases}$$

For Example: BIN metric assigned to the question “Does the system provide Spell Check feature” can be fully satisfied i.e. *TRUE* or fully declined *FALSE*.

A.2.2 Ordinal Metric (ORDN)

These are used for qualitative measurement. A metric x that ORDN-based is estimated on an ordinal scale, x ∈ { Zero Match, Very Poor Match, Poor match, Exact Match, Over Match}. Scale assumes the equal difference between the values.

$$\text{Value Given} = \begin{cases} 0 \text{ iff } x = \text{“Zero Match”} \\ 1 \text{ iff } x = \text{“Very Poor Match”} \\ 2 \text{ iff } x = \text{“Poor Match”} \\ 3 \text{ iff } x = \text{“Exact Match”} \\ 4 \text{ iff } x = \text{“Over Match”} \end{cases}$$

For Example: ORDN metric assigned to the question “Is the system have the Multi-View feature in order to have the comparison” can be evaluated using the ordinal scale.

Appendix B

GQM Based Feature Analysis

B.1 Overview

GQM, the acronym for "goal, question, metric", is an approach to software metrics that has been proposed by Dr. Victor Basili. In [9], According to Basili et al, GQM presents a systematic approach for integrating goals to models of the software processes, products and quality perspectives of interest based upon the specific needs of the project and the organization.

It provides a 3-step framework for the measurement of the organisational business:-

- The major goals of the system under development.
- Questions that define goals as completely in each perspective.
- Measurement that provide most appropriate information for answering the questions.

B.2 GQM Based Templates

As per the requirements identified in search phase, features are identified using GQM based template.

B.2.1 Functional Requirements

Table B.1: GQM Template for "Basic Editing Features"

Feature	Functional	
Sub-Feature	Basic Editing Features	
Description	The Features enable programmers to edit the document with much ease.	
Operational Questions		***Used Metric***
1. Does it offer a shortcut to complete an existing word (Auto complete)?		BIN
2. Does it offer line numbering feature?		BIN
3. Does the system provide Bookmark facility to the users?		BIN
4. Does the system support Drag-Drop feature?		BIN
5. Does the system support multiple undo/redo?		ORDN
6. Does the system able to search the contents with matching case?		BIN
7. Does the system support the regular expression search?		BIN
8. Does the system able to search the content in Files/folders/subdirectories?		BIN
9. Does the system have the WYSIWYG text editor?		BIN
10. Does the system is able to open the simultaneous file in tabs?		BIN
11. Does the system have the Zoom-in/ out facility?		BIN

Table B.2: GQM Template for "Programming Editing Features"

Feature	Functional	
Sub-Feature	Programming Editing Features	
Description	These features enable programmers to easily write programs in short interval of time & with less effort.	
Operational Questions		***Used Metric***
1. Does it offer a technique to highlight the syntax for various languages like C, C++, JAVA, PHP, Perl, and Python?		BIN
2. Is the system provides the Syntax Folding feature?		BIN
3. Does it offer Code comment/ uncomment feature?		BIN
4. Does the system provide compilation & execution of the scripts?		BIN
5. Does the system provide Bracket Matching?		BIN

Table B.3: GQM Template for "Richness of Built-in Tools"

Feature	Functional
Sub-Feature	Richness of Built-In tools

Description	The availability and variety of extra tools built in the system	
Operational Questions		***Used Metric***
1. Does it offer a feature in order to have the spell check of the document?		BIN
2. Does it offer a tool to have the statistics of the words present in the document?		ORDN
3. Is it able to mail the document as attachment/ text or export it?		ORDN
4. Is the system having the Multi-view feature in order to have the comparison?		ORDN
5. Does the system support UNICODE Encoding scheme?		BIN
6. Is the system having the session management feature?		BIN

B.2.2 Non-Functional Requirements

Table B.4: GQM Template for "Vendor's Support"

Feature	Non-Functional	
Sub-Feature	Vendor's Support	
Description	The trustworthiness of the available vendors.	
Operational Questions		***Used Metric***
1. Has the Vendor been good in past?		ORDN
2. How the market as a whole thinks about the product?		ORDN
3. A vendor that has been in business for a long time is more likely to continue providing upgrades and maintenance services in the future?		ORDN

Table B.5: GQM Template for "Help & support"

Feature	Non-Functional	
Sub-Feature	Help & Support	
Description	The level of support and help provided to the system users.	
Operational Questions		***Used Metric***
1. Does the system provide books and manuals for installation purpose?		ORDN
2. Does the system provide free support by expert?		ORDN
3. Does the system provide free manual & online help for application usage?		ORDN
4. Does the system provide free training, video, and mailing list help for application usage?		ORDN

Table B.6: GQM Template for "Usability"

Feature	Non-Functional	
Sub-Feature	Usability	
Description	The ease of use that the system provides to the end-users.	
Operational Questions		***Used Metric***
1. Does it support for Macro recording and playback facility to ease out the work?		BIN
2. Does it provide the feature for Char Case Change?		ORDN
3. Does the system provide feature for staying on top?		BIN

Table B.7: GQM Template for "Flexibility"

Feature	Non-Functional	
Sub-Feature	Flexibility	
Description	The extensibility of the system, contents, and users' information.	
Operational Questions		***Used Metric***
1. Is the user provided with option to change the Font size?		ORDN
2. Is the user provided with option to have different Font style and colour setting?		ORDN
3. Is the system having the plug-in feature in order to extend the functionality of the system?		BIN
4. Does the system provide Multilanguage Environment support like Chinese, Arabic etc...?		ORDN

Appendix C

Full Results of the Case Study

C.1 Snapshot of Most Promising Candidates

This appendix presents the various snapshots of the 5 candidates filtered from the 2nd phase Filtration with the highlighted features.

C.1.1 Crimson Editor

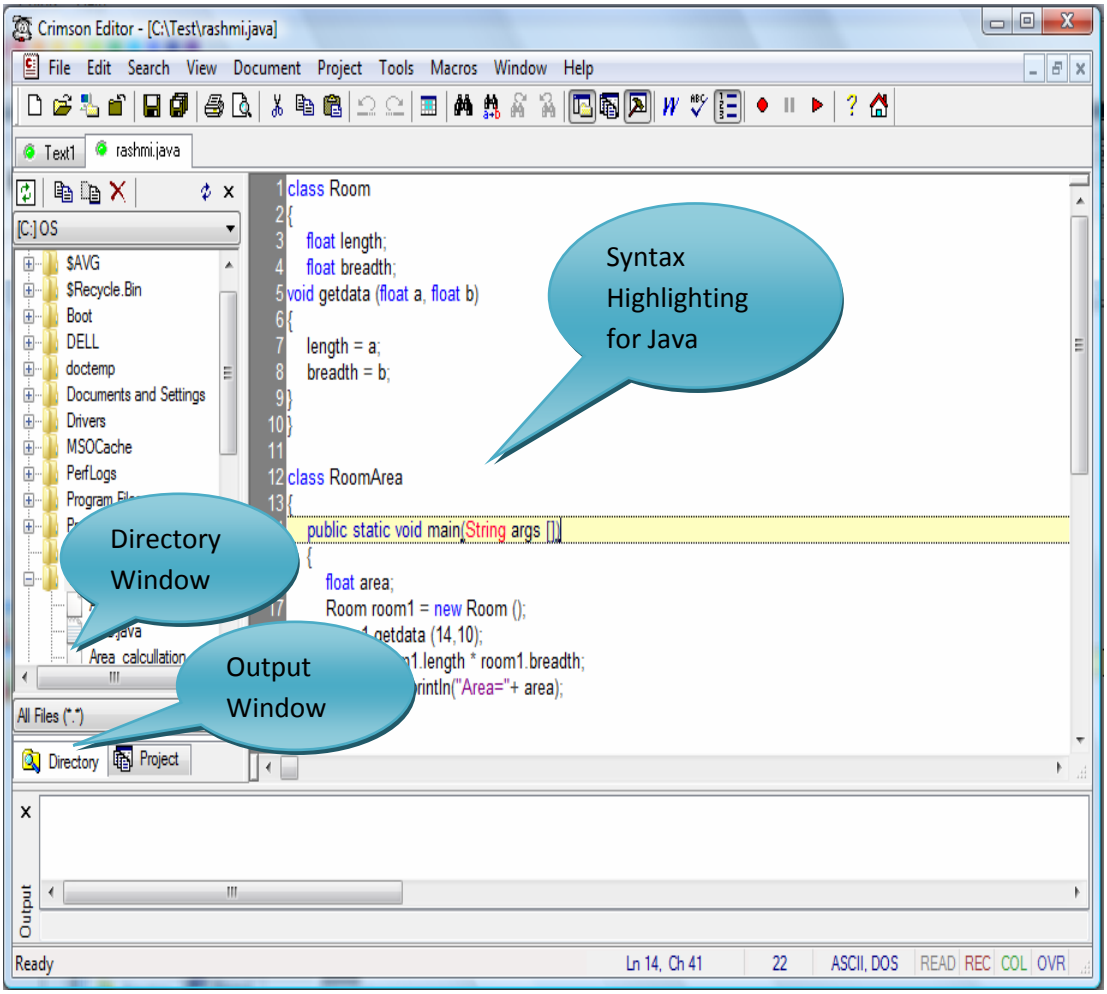

Figure C.1: Snapshot for Crimson Editor

C.1.2 Notepad++

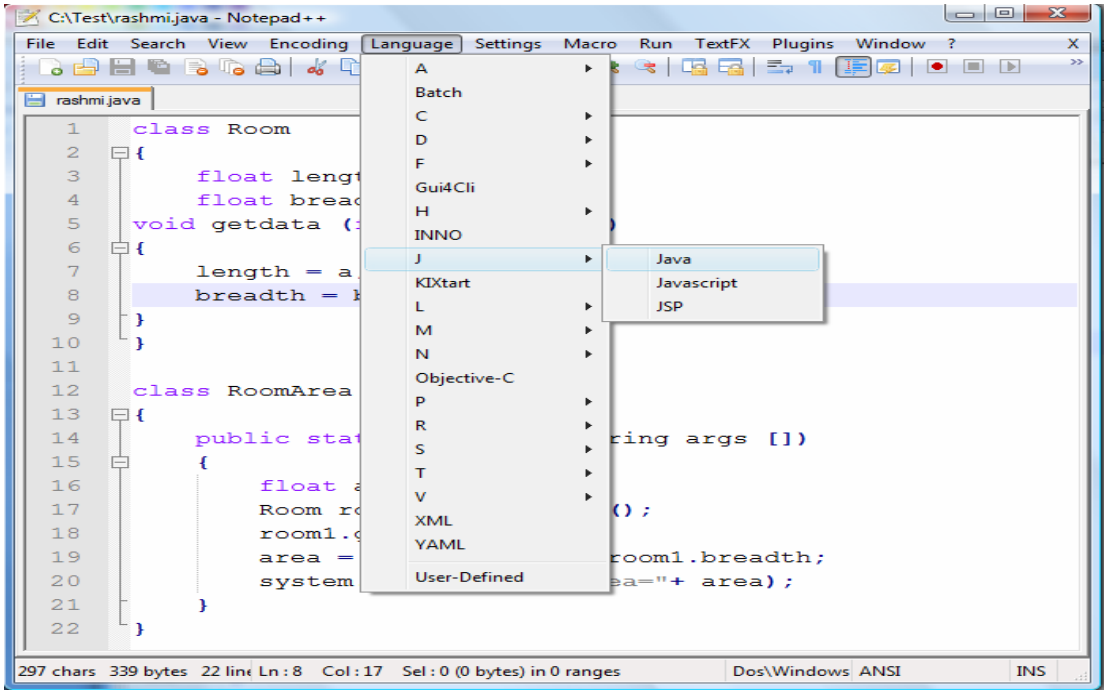

Figure C.2: Snapshot for Notepad++

C.1.3 Programmer's Notepad

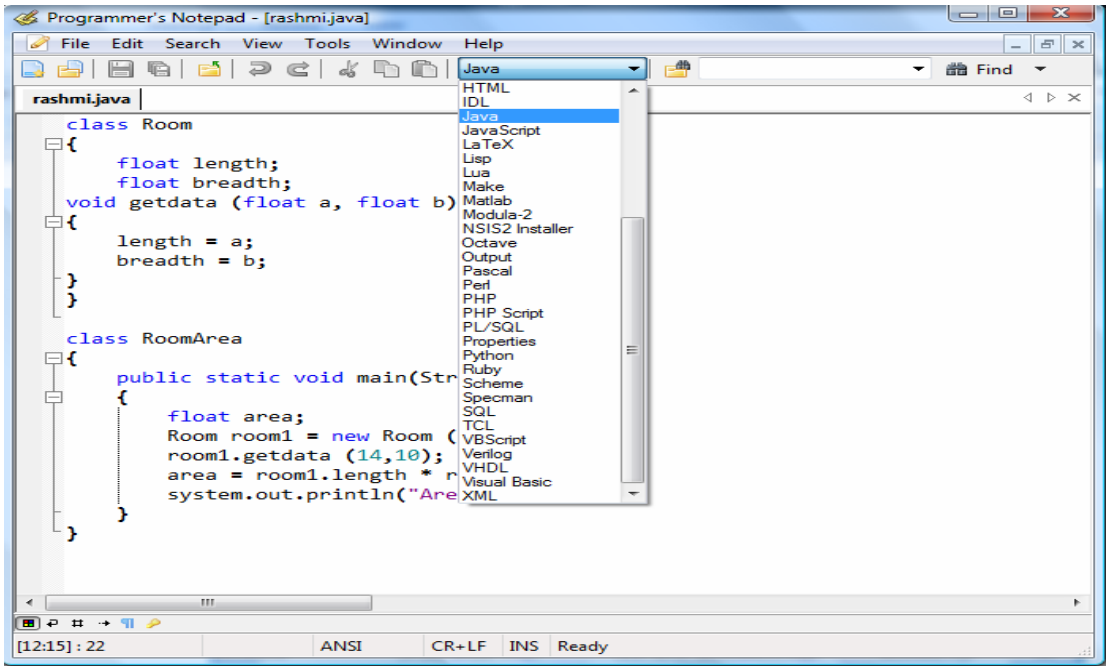

Figure C.3: Snapshot for Programmer's Notepad

C.1.4 Note tab Standard

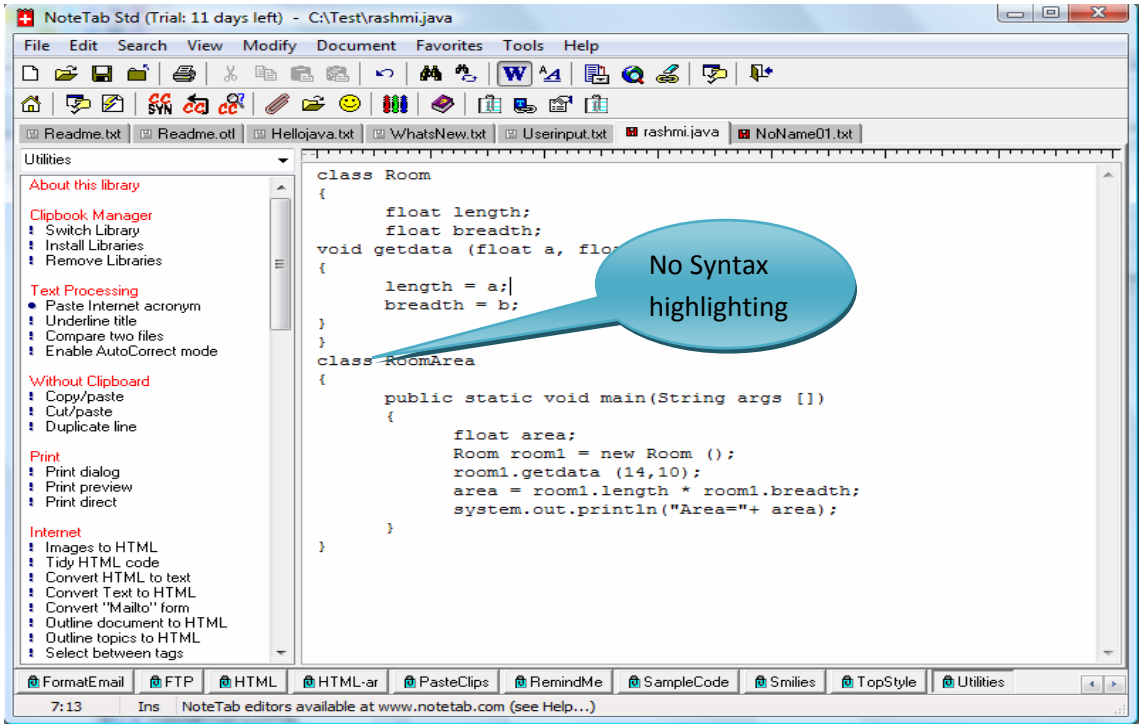

Figure C.4: Snapshot for Notetab Standard

C.1.5 PSPAD Editor

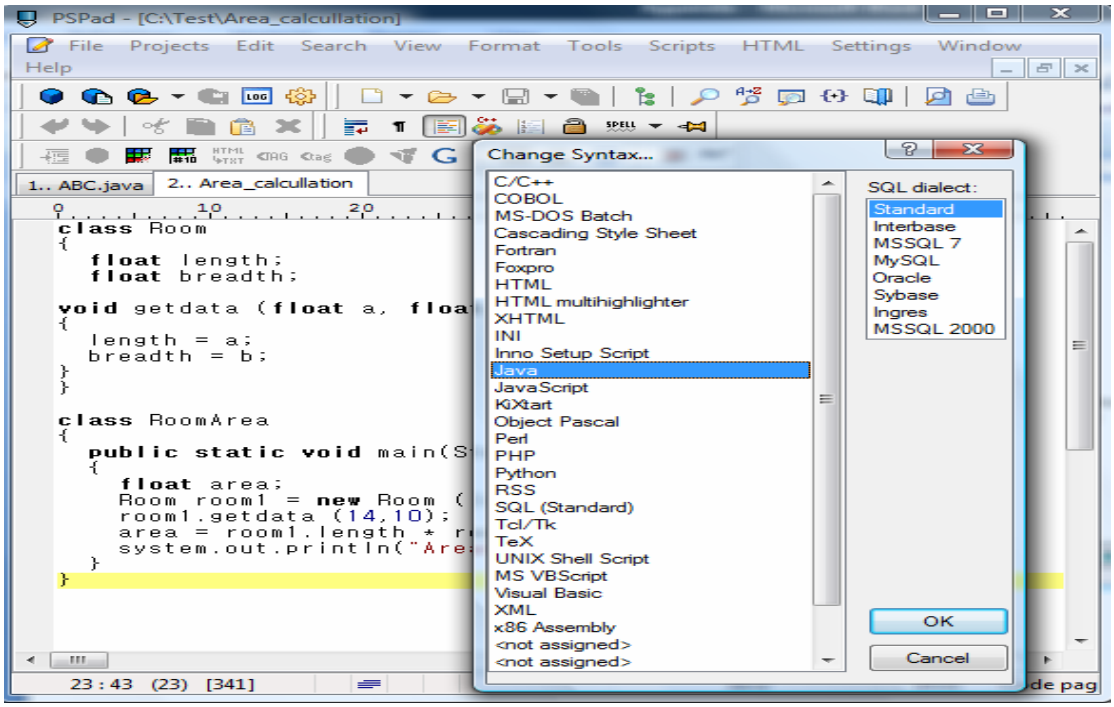

Figure C.5: Snapshot for PSPAD Editor

C.2 Feature Evaluation

This appendix presents the results of evaluation of five promising candidates using GQM template listed in appendix A.2. It results in total satisfaction value of the COTS components.

Features			COTS 2 (Crimson Editor)			COTS 7 (Notepad++)			COTS 11 (PN2)			COTS 9 (Notetab-Std)			COTS 12 (PSPAD)		
GQM Based Operational Questions		Metric Type	Value	Satisfaction Value	Type	Value	Satisfaction Value	Type	Value	Satisfaction Value	Type	Value	Satisfaction Value	Type	Value	Satisfaction Value	Type
Basic Editing Features	Does it offer a shortcut to complete an existing word (Autocomplete)?	BIN	0	0.00	FALSE	1	1.00	TRUE	1	1.00	TRUE	0	0.00	FALSE	1	1.00	TRUE
	Does it offer line numbering feature?	BIN	1	1.00	TRUE	1	1.00	TRUE	1	1.00	TRUE	0	0.00	FALSE	1	1.00	TRUE
	Does the system provide Bookmark facility to the users?	BIN	1	1.00	TRUE	1	1.00	TRUE	1	1.00	TRUE	0	0.00	FALSE	1	1.00	TRUE
	Does the system support Drag-Drop feature?	BIN	1	1.00	TRUE	1	1.00	TRUE	1	1.00	TRUE	1	1.00	TRUE	1	1.00	TRUE
	Does the system support multiple undo/redo?	ORDN	3	1.00	Exact Match	3	1.00	Exact Match	3	1.00	Exact Match	1	0.33	Very Poor Match	3	1.00	Exact Match
	Does the system able to search the contents with matching case?	BIN	1	1.00	TRUE	1	1.00	TRUE	1	1.00	TRUE	1	1.00	TRUE	1	1.00	TRUE
	Does the system support the regular expression search?	BIN	1	1.00	TRUE	1	1.00	TRUE	1	1.00	TRUE	1	1.00	TRUE	1	1.00	TRUE

Figure C.6: Feature Evaluation Results Part (1/6)

Features			COTS 2 (Crimson Editor)			COTS 7 (Notepad++)			COTS 11 (PN2)			COTS 9 (Notetab-Std)			COTS 12 (PSPAD)		
GQM Based Operational Questions		Metric Type	Value	Satisfaction Value	Type	Value	Satisfaction Value	Type	Value	Satisfaction Value	Type	Value	Satisfaction Value	Type	Value	Satisfaction Value	Type
	Does the system able to search the content in files/folders/subdirectorie?	BIN	1	1.00	TRUE	1	1.00	TRUE	1	1.00	TRUE	0	0.00	FALSE	1	1.00	TRUE
	Does the system have the WYSIWYG text editor?	BIN	1	1.00	TRUE	1	1.00	TRUE	1	1.00	TRUE	0	0.00	FALSE	1	1.00	TRUE
	Does the system is able to open the simultaneous file in tabs?	BIN	1	1.00	TRUE	1	1.00	TRUE	1	1.00	TRUE	1	1.00	TRUE	1	1.00	TRUE
	Does the system have the Zoom-in/ out facility?	BIN	0	0.00	FALSE	1	1.00	TRUE	1	1.00	TRUE	0	0.00	FALSE	0	0.00	FALSE
ming Editing Features	Does it offer a technique to highlight the syntax for various languages like C, C++, JAVA, PHP, Perl, Python?	BIN	1	1.00	TRUE	1	1.00	TRUE	1	1.00	TRUE	0	0.00	FALSE	1	1.00	TRUE
	Is the system provide the Syntax Folding feature?	BIN	0	0.00	FALSE	1	1.00	TRUE	1	1.00	TRUE	0	0.00	FALSE	0	0.00	FALSE
	Does it offer Code comment/ uncomment	BIN	0	0.00	FALSE	1	1.00	TRUE	1	1.00	TRUE	0	0.00	FALSE	1	1.00	TRUE

Figure C.6: Feature Evaluation Results Part (2/6)

Features			COTS 2 (Crimson Editor)			COTS 7 (Notepad++)			COTS 11 (PN2)			COTS 9 (Notetab-Std)			COTS 12 (PSPAD)		
GQM Based Operational Questions		Metric Type	Value	Satisfaction Value	Type	Value	Satisfaction Value	Type	Value	Satisfaction Value	Type	Value	Satisfaction Value	Type	Value	Satisfaction Value	Type
Program	Does the system provide compilation & execution of the scripts?	BIN	1	1.00	TRUE	0	0.00	FALSE	0	0.00	FALSE	0	0.00	FALSE	1	1.00	TRUE
	Does the system provide Bracket Matching?	BIN	0	0.00	FALSE	0	0.00	FALSE	0	0.00	FALSE	0	0.00	FALSE	1	1.00	TRUE
Richness of Built-In Tools	Does it offer a feature in order to have the spell check of the document?	BIN	0	0.00	FALSE	0	0.00	FALSE	0	0.00	FALSE	0	0.00	FALSE	1	1.00	TRUE
	Does it offer a tool to have the stastics of the words present in the document?	ORDN	3	1.00	Exact Match	1	0.33	Very Poor Match	0	0.00	Zero Match	3	1.00	Exact Match	1	0.33	Very Poor Match
	Is it able to mail the document as attachment/ text or export it?	ORDN	0	0.00	Zero Match	1	0.33	Very Poor Match	2	0.67	Poor Match	3	1.00	Exact Match	3	1.00	Exact Match
	Is the system having the Multi-view feature in order to have the comparison?	ORDN	0	0.00	Zero Match	1	0.33	Very Poor Match	1	0.33	Very Poor Match	0	0.00	Zero Match	3	1.00	Exact Match
	Does the system support UNICODE Encoding scheme?	BIN	1	1.00	TRUE	1	1.00	TRUE	1	1.00	TRUE	0	0.00	FALSE	1	1.00	TRUE

Figure C.6: Feature Evaluation Results Part (3/6)

Features			COTS 2 (Crimson Editor)			COTS 7 (Notepad++)			COTS 11 (PN2)			COTS 9 (Notetab-Std)			COTS 12 (PSPAD)		
GQM Based Operational Questions		Metric Type	Value	Satisfaction Value	Type	Value	Satisfaction Value	Type	Value	Satisfaction Value	Type	Value	Satisfaction Value	Type	Value	Satisfaction Value	Type
	Is the system having the session management feature?	BIN	0	0.00	FALSE	1	1.00	TRUE	0	0.00	FALSE	0	0.00	FALSE	1	1.00	TRUE
Vendor's Support	Has the Vendor been good in past?	ORDN	2	0.67	Poor Match	3	1.00	Exact Match	3	1.00	Exact Match	2	0.67	Poor Match	3	1.00	Exact Match
	How the market as a whole thinks about the product?	ORDN	1	0.33	Very Poor Match	3	1.00	Exact Match	3	1.00	Exact Match	2	0.67	Poor Match	3	1.00	Exact Match
	A vendor that has been in business for a long time is more likely to continue providing upgrades and maintenance services in the future?	ORDN	1	0.33	Very Poor Match	3	1.00	Exact Match	3	1.00	Exact Match	2	0.67	Poor Match	3	1.00	Exact Match
Support	Does the vendor provide books and manuals for installation purpose?	ORDN	0	0.00	Zero Match	0	0.00	Zero Match	2	0.67	Poor Match	0	0.00	Zero Match	2	0.67	Poor Match
	Does the vendor provide free support by expert?	ORDN	1	0.33	Very Poor Match	1	0.33	Very Poor Match	1	0.33	Very Poor Match	1	0.33	Very Poor Match	1	0.33	Very Poor Match

Figure C.6: Feature Evaluation Results Part (4/6)

Features			COTS 2 (Crimson Editor)			COTS 7 (Notepad++)			COTS 11 (PN2)			COTS 9 (Notetab-Std)			COTS 12 (PSPAD)		
GQM Based Operational Questions		Metric Type	Value	Satisfaction Value	Type	Value	Satisfaction Value	Type	Value	Satisfaction Value	Type	Value	Satisfaction Value	Type	Value	Satisfaction Value	Type
Help &	Does the vendor provide free manual & online help for application usage?	ORDN	3	1.00	Exact Match	3	1.00	Exact Match	3	1.00	Exact Match	3	1.00	Exact Match	3	1.00	Exact Match
	Does the vendor provide free training , video, mailing list help for application	ORDN	0	0.00	Zero Match	2	0.67	Poor Match	1	0.33	Very Poor Match	2	0.67	Poor Match	1	0.33	Very Poor Match
Usability	recording and playback facility to ease out the work?	BIN	1	1.00	TRUE	1	1.00	TRUE	1	1.00	TRUE	0	0.00	FALSE	1	1.00	TRUE
	Does it provide the feature for Char Case Change?	ORDN	3	1.00	Exact Match	2	0.67	Poor Match	2	0.67	Poor Match	3	1.00	Exact Match	3	1.00	Exact Match

Figure C.6: Feature Evaluation Results Part (5/6)

Features			COTS 2 (Crimson Editor)			COTS 7 (Notepad++)			COTS 11 (PN2)			COTS 9 (Notetab-Std)			COTS 12 (PSPAD)		
GQM Based Operational Questions		Metric Type	Value	Satisfaction Value	Type	Value	Satisfaction Value	Type	Value	Satisfaction Value	Type	Value	Satisfaction Value	Type	Value	Satisfaction Value	Type
	Does the system provide feature for staying on top?	BIN	1	1.00	TRUE	1	1.00	TRUE	1	1.00	TRUE	1	1.00	TRUE	1	1.00	TRUE
Flexibility	Is the user provided with option to change the Font size?	ORDN	3	1.00	Exact Match	3	1.00	Exact Match	3	1.00	Exact Match	3	1.00	Exact Match	3	1.00	Exact Match
	option to have different Font style and colour setting?	ORDN	3	1.00	Exact Match	3	1.00	Exact Match	3	1.00	Exact Match	3	1.00	Exact Match	3	1.00	Exact Match
	Does the system provide Multilanguage Environment support like Chinese, Arabic etc...?	ORDN	0	0.00	Zero Match	3	1.00	Exact Match	1	0.33	Very Poor Match	0	0.00	Zero Match	3	1.00	Exact Match
	Is the system having the plug-in feature in order to extend the functionality of the system?	BIN	0	0.00	FALSE	1	1.00	TRUE	1	1.00	TRUE	0	0.00	FALSE	0	0.00	FALSE
	Satisfaction (COTS)			20.67			28.67			27.33			14.33			30.67	

Figure C.6: Feature Evaluation Results Part (6/6)

C.3 Identified Mismatches and their Mitigation Action

This section lists the identified mismatches and for each mismatch, its mismatch value MV (x_i), description and new mismatch value after handling decision also included. It also lists the mitigation action like plug-in, refine requirements used to resolve the mismatches identified. After using the action, new mismatch value is calculated by

setting the mismatch value to 0 if the mismatch is completely resolved or ignored and remains the same if not resolvable.

C.3.1 Notepad ++

Features	COTS 7 (Notepad ++)			
GQM Based Operational Questions	Mismatch Value	Description	Mitigation Action	New Mismatch Value (MV)
Does it offer a feature in order to have the spell check of the document?	1.00	Feature not present	Resolve using Spell Check Plugin	0
Does it offer a tool to have the stastics of the words present in the document?	0.67	Tool counts the words/char/lines in the selected text	Resolve using Math pad plugin feature	0
Is it able to mail the document as attachment or text or export it?	0.67	It mails the document via outlook but not export it as HTML/RTF	Resolve by adding plugin for exporting file as RTF or HTML	0
Is the system having the Multi-view feature in order to have the comparison?	0.67	Earlier it supports Multi-view only but no highlighting of text differnecs	Resolve using Compare plugin feature	0
Does the system provide compilation & execution of the scripts?	1.00	Feature not present	Not Resolvable	1
Does the system provide Bracket Matching?	1.00	Feature not present	Refine using the plugin "Xbracket "which provides Autocompletion of brackets also	0
Does the vendor provide books and manuals for installation purpose?	1.00	Feature not present	Resolve by adding the Documentation for installation	0
Does the vendor provide free support by expert?	0.67	It provides only online help and contents help	Refine as the vendor provide help through Forum Discussion	0
Does the vendor provide free training , video, mailing list help for application usage?	0.33	It provides only online help and contents help	Ignore	0
Does it provide the feature for Char Case Change?	0.33	It provides change to Uppercase & Lowercase but not invertion & Capitalize feature	Ignore	0
Total	7.33			1

Figure C.7: Mismatch Handling for Notepad++

C.3.2 Programmer's Notepad

Features	Programmer's Notepad			New Mismatch Value (MV)
GQM Based Operational Questions	Mismatch Value (MV)	Description	Mitigation Action	
Does it offer a feature in order to have the spell check of the document?	1.00	Spell Check feature is absent here.	Resolve by adding SpellCheck plugin	0
Does it offer a tool to have the stastics of the words present in the document?	1.00	Tool to count the stastics of the characters/words/lines is absent	Resolve by adding Mathpad tool	0
Is it able to mail the document as attachment or text or export it?	0.33	Export the document as RTF/ HTML is there but not able to send the mail.	Not resolvable	0.33
Is the system having the Multi-view feature in order to have the comparison?	0.67	Multi-view(either horizontal/vertical is there) but not able to highlight the text diffrence	Not resolvable	0.666667
Is the system having the session management feature?	1.00	Session management for the active files is not present	Not resolvable	1
Does the system provide compilation & execution of the scripts?	1.00	Feature not present	Not resolvable	1
Does the system provide Bracket Matching?	1.00	Feature not present	Not resolvable	1
Does the vendor provide books and manuals for installation purpose?	0.33	Feature not present	Resolve by adding the Documentation for installation	0
Does the vendor provide free support by expert?	0.67	it provides online help and contents help only	Refine as the vendor provide help through Forum Discussion	0
Does the vendor provide free training , video, mailing list help for application usage?	0.67	it provides online help and contents help only	Ignore	0
Does it provide the feature for Char Case Change?	0.33	It provides change to Uppercase & Lowercase but not invertion, Capitalize feature	Ignore	0
TOTAL	8.00			3.996667

Figure C.8: Mismatch Handling for Programmer's Notepad

C.3.3 PSPAD Editor

Features	PSPAD Editor			New
GQM Based Operational Questions	Mismatch Value (MV)	Description	Mitigation Action	Mismatch Value (MV)
Does it offer a tool to have the stastics of the words present in the document?	0.67	Tool count the words/char/lines at the time text selection	Not Resolvable	0.67
Does the system have the Zoom-in/ out facility?	1.00	Feature not present	Refine as it provides to increase the Font Size	0
Is the system provide the Syntax Folding feature?	1.00	Feature not present	Not Resolvable	1
Does the vendor provide books and manuals for installation purpose?	0.33	Feature not present	Resolve by adding the Documentation for installation	0
Does the vendor provide free support by expert?	0.67	it provides online help and contents help only	Refine as the vendor provide help through Forum Discussion	0
Does the vendor provide free training , video, mailing list help for application usage?	0.67	it provides online help and contents help only	Ignore	0
Is the system having the plug-in feature in order to extend the functionality of the system?	1.00	Feature not present	Not Resolvable	1
Total	5.33			2.67

Figure C.9: Mismatch Handling for PSPAD Editor