Toward the use of Relative Benefit For Case Base Maintenance In Electronic shop

by

Wacharee Ditcharoen

Submitted in Partial Fulfillment of
the Requirements for the Degree of
Master of Science
in Information Technology
Assumption University

May, 2002

Toward the use of Relative Benefit For Case Base Maintenance In Electronic shop

By

Wacharee Ditcharoen

Submitted in Partial Fulfillment of the Requirements for the Degree of Master of Science in Information Technology Assumption University

May, 2002

The Faculty of Science and Technology

Thesis Approval

Thesis Title	Toward the use of Relative Benefit For Case Base Maintenance In Electronic shop
By	Ms. Wacharee Ditcharoen
Thesis Advisor	Dr. Jirapun Daengdej
Academic Year	3/2001

The Department of Information Technology, Faculty of Science and Technology of Assumption University has approved this final report of the **twelve** credits course. **IT7000 Master Thesis**, submitted in partial fulfillment of the requirements for the degree of Master of Science in Information Technology.

Approval Committee:

(Dr. Jirapun Daengdej)
Advisor

(Asst.Prof.Dr. Thotsapon Sortrakul)
Committee Member

(Dr. Thiraphong Charoenkhunwiwat)
Committee Member

(Professor Dr. Chidchanok Lursinsap)
Representative of Ministry of University Affairs

Faculty Approval:

(Asst.Prof.Dr. Thotsapon Sortrakul)
Director

(Asst.Prof.Dr. Pratit Santiprabhob)
Dean

June/ 2002

ACKNOWLEDGEMENTS

I sincerely thank my advisor, Dr. Jirapun Daengdej, for his valuable guidance, and advice throughout this study. I would also like to thank the other members of my committee, Asst. Prof. Dr. Thotsapon Sortrakul and Dr. Thiraphong Charoenkhunwiwat for their guidance and support. Thank to Khun Sorapong for his suggestion of mathematic. I also thank Khun Anan for providing useful resources especially on Microsoft SQL Server.

Finally, thank to my parents, the other members of the family, and my friends for supporting me all along the way.

ABSTRACT

In today's environment where the electronic commerce becomes more and more available and important, a case-based reasoning (CBR) is one of the widely used techniques in product searching and sales selection process. As it becomes more popular, the library scale of the CBR system is also growing. Theoretically, large scale of library will impact the overall efficiency of the system in term of searching, retrieving data as well as storing new cases.

As a result, a maintenance so called case-based maintenance is required in order to prevent any effect from having large database. However, the reducing amount of case library must be done carefully in order not to produce any damage to the system's objectives and overall accuracy. Some previous researches usually maintain the case library by focusing only either on the performance or the competency of the case-based system. But in electronic shop, removal of case library needs to concern more than just performance. The other factors, such as the frequency of sell, the availability of the product if that case represent product existing in the stock, should also be taken into account when maintaining the case library.

Deletion method should minimize the size of the case-based while maintaining the benefits (objectives) of the domain (electronic shop). This thesis aims to propose the deletion method, where the benefit of the domain will not be traded off. The proposed deletion method will be done by taking all necessary factors into consideration and modeling the relative benefit value, which is mainly guided in deleting the case library.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TALBES

CHAPTER1: INTRODUCTION

1.1 Background

Case-Based Reasoning (CBR) is widely used for the search and the selection process in the sales system. To date in electronic shop, the sales system allows customers enable interactive to the system to find appropriated products for meeting customer satisfaction [2,3]. The example of such system, i.e.WEBSELL [2], is designed by applying CBR technology for product selection and customization in electronic commerce environments. The growing access and the use of the system lead to the huge portions of Case-Based. An uncontrolled case-base growth can cause serious performance problems as the retrieval efficient degradation and incorrect or inconsistent cases which becomes increasingly difficult to detect [1]. At the same time, configurations of products are frequently changed (e.g., price, new hardware components, etc.) [2]. Therefore, the case-based system must be controlled and maintained to support the incremental up-date of changing.

1.2 Problem statement

As Case-Based Reasoning is widely used in problem solving and various applications especially in Electronic Commerce, the growing use of CBR system leads to the large scale of case library. The large case base without monitor, control, or reducing amount of case is the cause of many problems in searching, retrieving, adaptation, or storing a new case in case library, especially, the incremental up-date of dynamic cases in electronic shop. The case base of base product needs to be monitored, controlled, and maintained to support the huge of business transaction. There are many researches that have been focused on the competence of CBR system but a few to describe its maintenance. Some research describes the competence-preserving case deletion policy. However, pure reduction case without analyzing the benefit for the commercial is very dangerous in the domain of Electronic Commerce. The loss of chance in business may occur during transactions causing a loss of some cases. The useful product (a case) might be deleted from the system without consideration of benefit viewpoint. These things lead to the loss of benefit for the electronic shop domain.

This thesis will present the deletion method to maintain the growing of case library that used in electronic shop by defining the relative benefit value to measure the quality of a case, and then use its analysis for selection case in adding or deletion method. The model is designed to facilitate, monitor, and control the incremental up-date of dynamic case-based.

1.3 Objectives

As maintenance case base reasoning applied in electronic shop, this thesis would present the strategy to maintain the growing of case library as follows these objectives:

1. Propose an appropriate technique of maintenance for the large case library used in electronic shop.
2. Define the relative benefit value to guide the case and use it to maintain the case in case library.
3. Create an algorithm to find out the relative benefit value.
4. Design a deletion model and its process for case reduction.
5. Simulate a prototype of adding case to the system. Demonstrate deletion technique to compare and see the result of remained cases in case library with other technique.

1.4 Scope and limitation of study

Case based reasoning is applied in various areas such as information retrieval, technical support or help desk, finance, engineering, www application, etc. The growing access and the use of the system lead to the huge portions of case library. Maintenance is very important to control amount of case and performance of case based system. The selection of appropriate technique to maintenance is also important. Selecting inappropriate technique leads to damage function of work and also loss of opportunities in such area. Electronic commerce or the area that used the case to represent either product is the area that must be carefully considered. This thesis is narrowly scoped to find out the deletion method to use for maintaining case library used in sales support system for electronic commerce application. The working area is scoped as follows:

- Research and learning CBR and survey the application or working area where CBR was applied to use.
- Survey and literate the previous approach of case base maintenance and then analyze advantage and weak point that has an impact to electronic commerce area.
- Find out an appropriate technique of maintenance for the large case library used in electronic shop.
- Definite relative benefit for each case and use it for adding or deletion case
- Simulate prototype to support algorithm by assumption the existing case library.

- To experiment maintenance by deletion case from the system following the process and deletion policy
- See the result of remained case in the case library. Compare the expected result with other technique.

CHAPTER 2: LITERATURE REVIEW

2.1 Introduction of Case based reasoning and product customized in Electronic shop

Kolodner [4] described the meaning of Case-based reasoning that is a method for solving the problem by remembering previous similar situations and reusing information and knowledge about that situation. Riesbeck and Schank [5] gave a simple idea that a case-based reasoner solves new problems by adapting solutions that were used to solve old problems. The case structure is shown in Figure 2-1. The case (d) is the solution of the adapted case (b) and case (y) in order of CBR cycle.

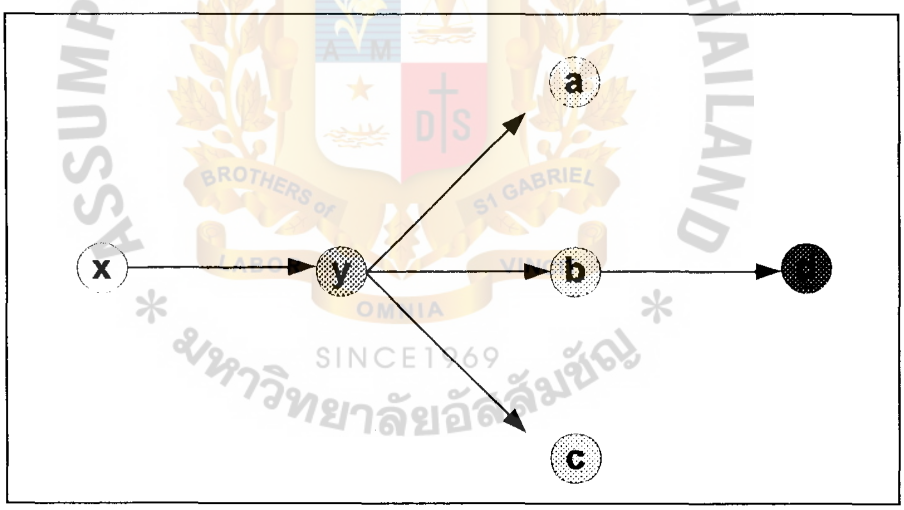


Figure 2-1: Case Base Structure Graph

Nowadays, case base reasoning is applied in various solutions to solve the problem. Electronic commerce is one domain where CBR is usually used. The first example is negotiation during intelligent sales support with case-base reasoning [6]. It presents architectures of sales agents, which are able to negotiate with a customer.

The other examples, Armin, Ralph, and Sascha suggest CBR technique for product selection and customization-structured product in Electronic shops [2,3]. Normally, most electronic catalogs and online shops do not explore the interactivity available on the Web, so they propose the interactive approach to customize products for many electronic commerce applications. They gave the configuration of PC domain as the example. Each case represents the description of the real product in the base. Very briefly, the configuration process can be subdivided into two major steps, called base product retrieval and adaptation cycle. The task of the first step is the similarity-based retrieval of the best available base product from the respective case-base. The second step is an iterative procedure that performs the necessary adaptation of the retrieved-base product if it does not fulfill all customer demands. The result of adaptation cycle is validated product. The complete configuration process is succeeded, and the final product can be presented to the customer as show in Figure 2-2.

In electronic commerce, the case contained in case library represents the real product. Maintenance case is very important to control and improve the competence of case base system. This is strictly careful to add, update, or delete a case in case base system.

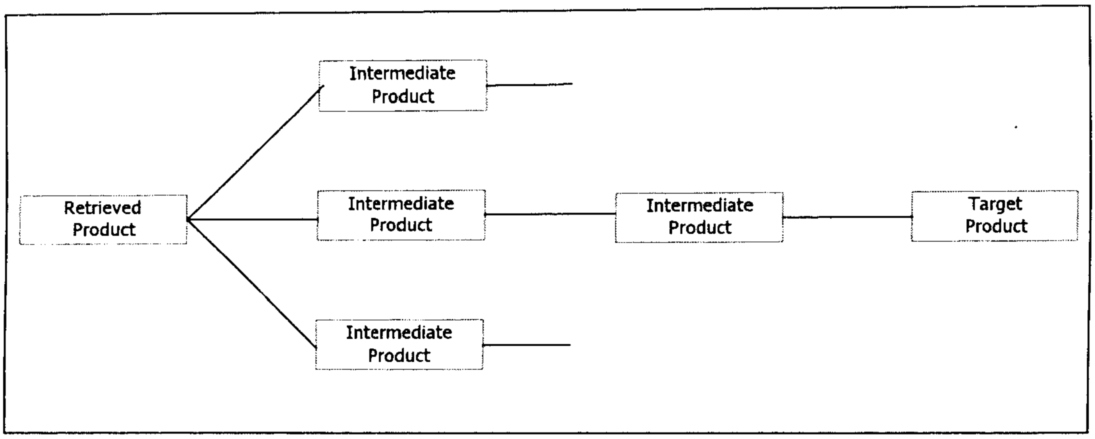


Figure 2-2: Customization of product by using CBR process

This paper is giving the exemplary domain of PC configuration, which can be easily understood. Imagine an only catalog of an electronic shop. Every single product, i.e., PC workstation or server in the present domain example, is represented by a case in case base system. As an example, such a case could have the following description:

The example of case in configuration PC

```
Case1=( "IBM",           /* brand */
        "Server",        /* type */
        "PentuemIII",    /* CPU */
        256,             /* memory */
        17,              /* monitor */
        56,              /* modem */
        10,              /* hard disk */
        70,000,          /* price */
        "IS256",         /* product code */
        "13/11/02",      /* End of date */
        ...)
```

The example of case in catalog of travel agency

```
Case2=( "Caribbean",     /* region */
        "Winter",        /* season */
        "Airplane",      /* transportation */
        10,              /* duration */
        1999.99,         /* price */
        "Beach",         /* holiday type */
```

1508, /* journey code */
"HolidayFlat", /* accommodation */
...)

Figure 2-3: Example of case in real-world application

In electronic commerce, the cases are descriptions of products. The problem description is a specification of a single product and possible demands the product can satisfy. The solution to the problem is an unambiguous reference to the product [2]. For configuration products such as computers or travel packages, the solution is not only the part number, but also possibly the entire configuration as shown in Figure 2-4.

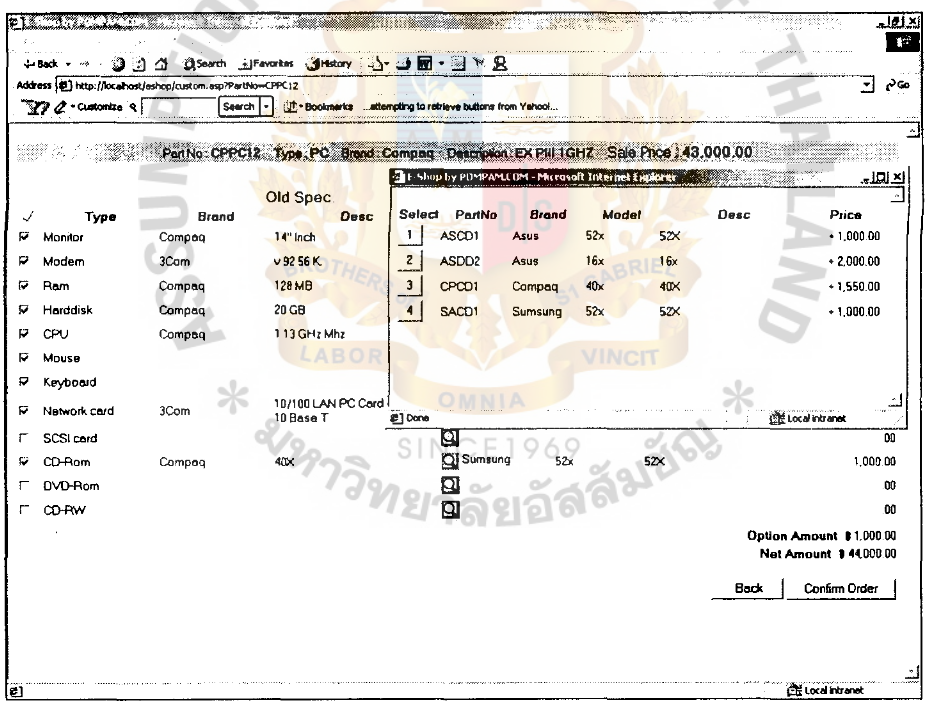


Figure 2-4: Product configuration in Electronic shop

In configuration of PC/Server, there are two types of case found, base product and custom made. Base product is a case that represents the original product that can be retrieved while custom made or intermediated product is a case that is modified and configured in learning process.

2.2 Previous approach of Case Base Maintenance

The previous research in maintaining case-base systems has addressed many different aspects of the direct reduction cases as follows:

2.2.1 Utility metric for deletion policy

Minton's utility metric [1,7,9] is a simple random deletion policy. A random item is removed from the knowledge-based one to the knowledge-base size that exceeds some predefined limited.

$$Utility = (ApplicationFreq * AverageSavings) - MatchCost \qquad (2.1)$$

The *utility metric*, which takes into account the cost of maintaining, is defined the knowledge item (retrieval or match cost) and expected the problem solving saving offered by the item (*average savings multiplied by application frequency*) [1], [9].

This technique chooses a knowledge item for deletion based on an estimate of its performance benefits. This utility deletion policy removes knowledge items with negative utility. This technique can work very well and can often be as effective as more principled and expensive methods [7]

- **Advantage**
  1) Work very well to reduce big amount of case from case library
  2) Easy to select the case considering only the negative utility for deletion

- **Drawback for electronic commerce**
  1) In electronic commerce, the retrieval time in www application should depend on the speed of internet linked and system performance of remote client in customer side while accessing the web. It's difficult to calculate the retrieval from customer side.
  2) The case that has negative utility will be removed from the system without other considerations. The case that represented the product that usually sell out or retrieve for solving problems might be deleted.

2.2.2 Competent preservation

Smyth and Keane [9] suggested a competence-preserving case deletion policy called the footprint deletion policy. This approach uses statistical techniques to calculate case competence. Each case should be classified according to its competence. The key concepts in categorizing cases are *coverage* and *reachability*. Coverage refers to the set of problems that each case can solve. Reachability is the set of case that can provide solutions for each current problem.

$$\text{Coverage}(c) = \{c' \in CB: \text{Adaptable}(c, c')\} \quad (2.2)$$

$$\text{Reachable}(c) = \{c' \in CB: \text{Adaptable}(c', c)\} \quad (2.3)$$

Using coverage and reachability, a case is *pivotal* if it is reachable by no other case but itself. *Auxiliary* case is the case that is completely subsumed by other cases in the case base. *Spanning cases* are the case that its coverage spaces link covered by other cases. *Support cases* exist in groups, each support providing similar coverage as others in a group. The deletion algorithm then deletes cases in the order of their classifications: auxiliary, support, spanning, and pivotal cases.

- **Advantage**
  1) This technique can help to model the competence of case base system while the model can exploit to against the competent deletion for controlling case base size.
- **Drawback for electronic commerce**
  1) It is typically difficult to calculate the actual coverage and reachability of a case because the possible set of problems is normally too extensive [6]. Thus it is assumed that the problem distribution in the case base is representative and a heuristic is used for further considerations. In electronic shop, the set of problem

exactly come from several user requirements that are very extensive. It is quite difficult to assume and uniform the requirement or expectation of various customers from the side of system administrator.

2) The competent case might not be the case that be of benefit for sales transaction in electronic commerce

2.2. 3 Identifying competence-critical instances

Brighton and Mellish introduce an algorithm, which they term the Iterative Filtering Algorithm (ICF) [14]. The ICF algorithm uses the instance-based learning parallels of case coverage and reachability. They applied a rule, which identifies cases that should be deleted. The reachable set is not fixed in size but rather bounded by nearest case of different class. They remove cases, which have a reachable set size greater than the coverage set size. A more intuitive reading of this rule is that a case "c" is removed when more cases can solve "c" than "c" can solve itself.

```
ICF(T)
    > Perform Wilson Editing
    for all x ∈ T do
        if x classified incorrectly by k nearest neighbors then
            flag x for removal
    for all x ∈ T do
        if x flagged for removal then T = T-{x}
    > Iterate until no cases flagged for removal
    repeat
        for all x ∈ T do
            compute reachable (x)
            compute coverage (x)
        progress = fals
        for all x ∈ T do
            if |reachable (x)| > |coverage (x)| then
                flag x for removal
```

```
                    progress = true
          for all x ∈ T do
                if x flagged for removal then T = T-{x}
until not progress
return T
```

Figure 2-5: The Iterative Case Filtering Algorithm

This is the deletion criterion the algorithm uses; the algorithm proceeds by repeatedly computing these properties after filtering has occurred. Usually, additional cases will begin to fulfill the criteria as thinning proceeds and the bands surrounding the class boundaries narrow. After a few iterations of removing cases and re-computing the sets, the criterion no longer holds.

- **Advantage**

  Above point turns out to be a very good point to stop removing cases as removing more cases tends to breach their objective of intrusive storage reduction.

- **Drawback for electronic commerce**

  1) Although this approach can preserve removing more case, the calculation of the actual coverage and reachability of a case were being a problem as the originality of Smyth and Keane's approach [9].
  2) The case that represented the product that usually sold out or retrieved for solving the problem will be deleted without other consideration except when applying the rule for deletion policy in competent case.

2.2.4 Performance-based metric

Leake and Wilson describe a strategy for performance-based case selection [13]. They used the relative performance (RP) metric aimed at assessing the contribution of a case to the adaptation performance of the system. The metric can be used to guide either case addition-favoring cases with high RP values—or case deletion—favoring cases with low RP values. They let RS(c',c) stand for ReachabilitySet(c')-{c}, for a fixed case-base the define:

$$RP(c) = \sum_{c' \in CovergeSet(c)} 1 - \frac{AdaptCost(c,c')}{Max_{c'' \subset RS(c',c)} AdaptCost(c'',c')} \qquad (2.4)$$

Suppose that if case C_1 solves problem p_1, the cost to adapt C_1 to solve new problem p_2 is $\alpha|p_1-p_2|$, for some fixed $\alpha > 0$

- **Advantage**

  This technique aims to adapt performance to case base system.

- **Drawback for electronic commerce**

  1) This technique intends to delete the case concerning relative performance of adaptation only.
  2) Although the remained case has a high performance of adaptation, it will not assure that the case usually retrieved and selected for sale in sales transaction may be removed.

2.3 Remained issue

The pure reduction case from the system is very dangerous for sales transaction. The above techniques work very well in competence preservation and performance based selection. However, in electronic shop, deletion each case effects to actual product in the shop. The error may occur in CBR cycle or sales transaction. There are many factors to peruse which case should be deleted from the base. Delete uncorrected case will cause a loss of benefit to domain (electronic shop).

2.4 Fuzzy logic for decision-making

Fuzzy systems can be used for estimating, decision-making, and mechanical control systems such as air conditioning, automobile controls, and even "smart" houses, as well as industrial process controllers and a host of other application.

Fuzzy decision-making is a specialized, language oriented fuzzy system used to make personal and business management decisions, such as purchasing cars and appliances [15].

2.4.1 Discovering fuzziness

Lotfi Zadeh introduced fuzzy logic in 1965. The basic idea was to extend the classical logic (the Boolean logic) in order to relax the harsh constraint that everything that can be said about anything is either absolutely true or absolutely false. Zadeh combines the concepts of crisp logic and the Lukasiewicz sets by defining graded membership. With fuzzy logic the answer is maybe, and its value ranges anywhere from

0 (No) to 1 (Yes). One of Zadeh's main insights was that mathematics can be used to link language and human intelligence.

2.4.2 Fuzzy set

Fuzzy set theory differs from classical set theory in on crucial aspect: An element can belong to the fuzzy set, be completely excluded from fuzzy set, or it can belong to the fuzzy set to any intermediate degree between these two extremes. The extend to which an element belongs to a given fuzzy set is called the grade of membership or degree of membership. The term of fuzzy was introduced by Zadeh to describe sets whose membership criteria are vague. Thus, small is only margininally a member of set of sales margin. Uncertainty about a statement such as the sales margin is small is not represented by the probability that the margin is small, but rather by Zadeh calls the possibility that the margin is small. The term small, medium, and high are imprecise terms. The inputs of a term are real-valued that people tend to respond that values by labeling them with a group name. For example, 10% of sales margin might be considered small margin of a sales. Real-valued attributes can present a problem for an expert system, because each value cannot be dealt with individually. Therefore, the values must be grouped together in some way. The possibility of a statement is represented by a number generated by a membership function. The membership function χ associated with a fuzzy set assigns degrees of membership to elements in the set. For example, we can assign the following values to the membership function associated with the margin of sales: the degree of membership of 10% margin in fuzzy set of small margins is 1 ($\chi[10] = 1$), but the degree of membership of 18% margin is only 0.2 ($\chi[18] = 0.2$).

A fuzzy set can also be represented by a quadratic equation (involving squares, n2, or numbers to the second power), which produces a continuous curve. Three shapes are possible, named for their appearance—the S function, the pi function, and the Z function [15],[16],[18]

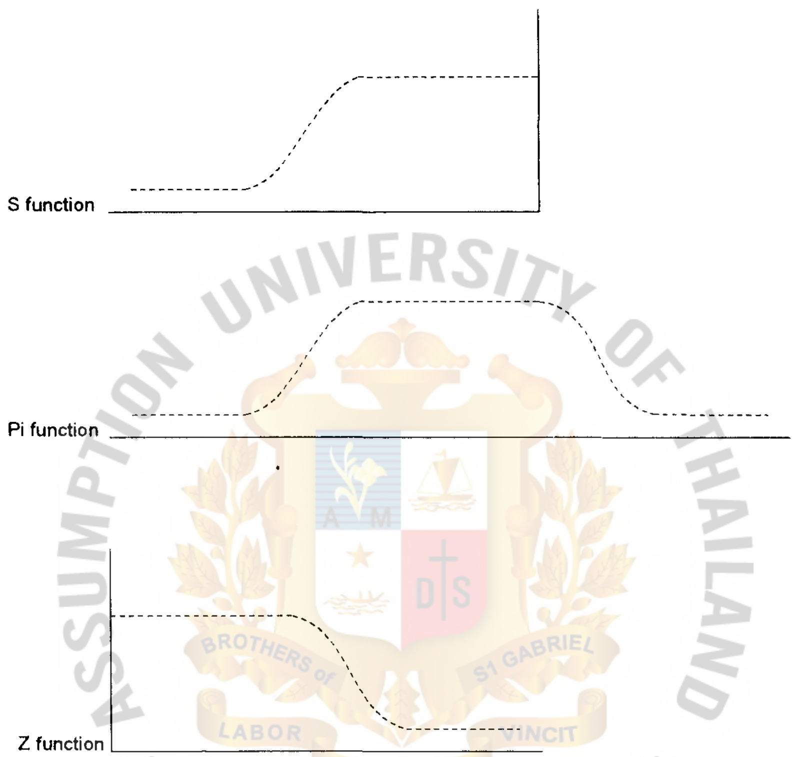


Figure 2-5: Graphs of the S function, the pi function, and the Z function

The degree of membership function of each sharp can be formulated from these following functions

$$\chi_A(X) \quad 1\text{-}S(x;a,b,c) = \begin{cases} 1 & \text{for } x \le a \\ 1-2((x-a)/(c-a))^2 & \text{for } a<x<b \\ 2((x-c)/(c-a))^2 & \text{for } b<x<c \\ 0 & \text{for } x \ge c \end{cases} \qquad (2.5)$$

where "a" and "c" are the function's end points, and b = (a+c)/2

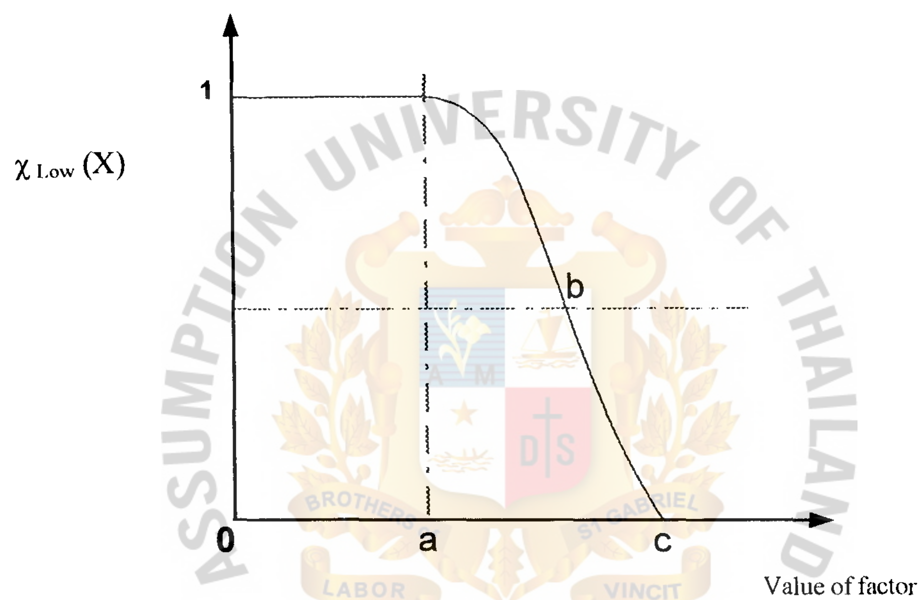


Figure 2-6: Membership function of LOW

A normal set, such as let A is the set of frequency of sell, is described by a characteristic function

$$\chi_A: X \in \{1,0\} \qquad (2.6)$$

In other word, if element x of X is included in A, it is expressed by $\chi_A(x) = 1$, and if not by $\chi_A(x) = 0$. The extension of the rang of this characteristic function, {0,1}, to the real number interval [0,1] gives the fuzzy set, and fuzzy set A of X is the set characterized by the membership function

$$\mu_A: X \rightarrow [0,1] \tag{2.7}$$

$\mu_A(x) \in [0,1]$ expresses the degree of membership of element x of X in A. Since {0,1} is included in rang [0,1] of μ_A, χ_A is a special case of μ_A, and it can be said that conventional sets are a special case of fuzzy sets. It is assumed that μ_A is determined based on human subjectivity.

Consider for example the fuzzy set LOW. The elements of the set are real values of each evaluation items whose membership grades depend on their results. For example, let the result of price performance be 10% might have a membership degree of 1, the result of price performance is 50% which might have a membership degree of 0, and price performance of intermediate LOW have intermediate grades of membership between 0 and 1. The fuzzy set LOW can be symbolized in the following way:

$$\text{LOW: result value of each evaluation item} \rightarrow [0,1] \tag{2.8}$$

Or degree of membership function of element x of X in Low can be expressed by

$$\mu_{Low}(x) \in [0,1] \tag{2.9}$$

The set can be represented in graphical form as shown in Figure 3-6. Let "0" to "c" is the membership of set LOW. If "a" is the highest value of evaluation item that can be accounted for degree of membership of set LOW is 1, the value between a and c need to interpolate a value for the grade of membership by using a smoothing function, called the 1-S-function or the other name called Z-function, which is defined as shown in Figure

2-6. The membership functions of fuzzy set A in the universe labeled LOW can be defined in the term of 1-S function.

Let B represent fuzzy set labeled HIGH, x is the real value of the result that get from evaluation item, the membership function of fuzzy set B can define the term of S-function [16] as defined in formula (2.10)

$$\chi_B(X) \quad S(x;a,b,c) = \begin{cases} 0 & \text{for } x \leq a \\ 2((x-a)/(c-a))^2 & \text{for } a<x<b \\ 1-2((x-c)/(c-a))^2 & \text{for } b<x<c \\ 1 & \text{for } x \geq c \end{cases} \qquad (2.10)$$

For the fuzzy set of MEDIUM, the membership function has a "bell shape" and is referred to as the π-function[16]:

$$\pi(x;b,c) = \begin{cases} S(x;\ c-b, c-b/2,\ c) & \text{for } x \leq c \\ 1-S(x;\ c,\ c+b/2, b+c & \text{for } x > c \end{cases} \qquad (2.11)$$

where S(x: .,   .,   .) is the S-function given in (2.10). In π-function, the parameter "c" is the point at which π is unity, and the parameter "b" represents the distance between the two crossover point of π (resulting from the two S-functions, and 1-S) and is referred to as the bandwidth of π -function.

CHAPTER 3: RELETIVE BENEFIT MODEL

3.1 Why Relative Benefit Model

Since, some cases contribute mainly to the competence of the system and others may predominantly contribute to its performance, the previous approaches have direct refection to real world application in Electronic commerce. They may not guarantee that the deleted case contributes mainly benefit or still is usually used or still be represents product being sold in the shop.

This section describes the new maintenance strategy, which is relative benefit model, to maintain case base system based on Relative Benefit value (RB) of each case. This model is designed to minimize the loss of necessary cases for the selling process.

The key concept is identifying relative benefit value and classification cases into low benefit group, medium benefit group, and high benefit group by taking necessary factors from case attribute and transaction of sales such as an average frequency of retrieval, frequency of sell , margin, and remaining day that case is available in the system to perform relative benefit value for measuring cases.

3.2 Introduction to Relative Benefit Structured

The relative benefit model consists of two major levels in structural. Identification of relative benefit value (RB) to each case is the first level. The second level is the selection of deletion policy.

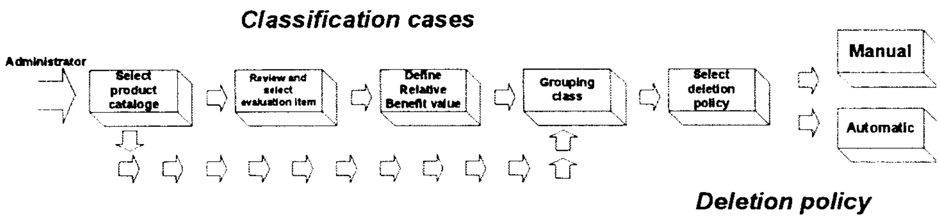


Figure 3-1: Structure of Relative Benefit Model

The cases are grouped into low benefit group, medium benefit group, and high benefit group depending on their relative benefit value that fit in each group. The cases grouped in the low benefit group will be deleted from the system before the cases grouped in the medium benefit group and the high benefit group in ordered. However, the cases with immediate end of life cycle will be the first priority to be selected for deletion without considering their relative benefit value.

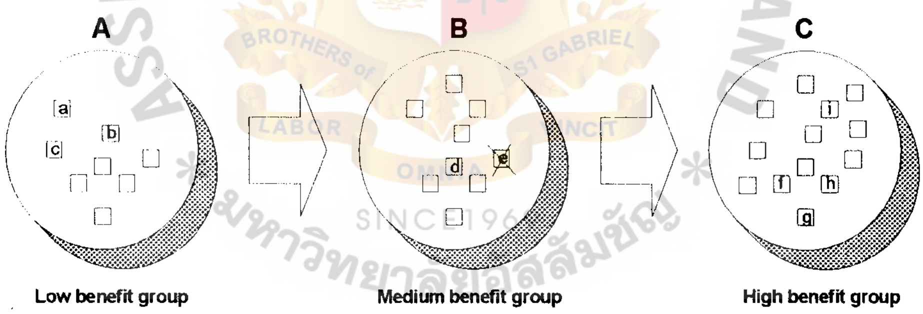


Figure 3-2: Case categories

Figure 3-2 illustrates the difference case categories in term of their benefit for case library and electronic shop. Each case is identified benefit value and modeled into

Low benefit group, Medium benefit group, and High benefit group in ordered with subjected by human depending on benefit value from less to much. Case "a", "b", "c" are classified into Low benefit cases which is the first group that will be selected to delete from the system before other while case "d" and "e" are grouped into medium benefit group. Case "f", "g", "h", and "i" are high beneficial cases, which are classified as high benefit group that are selected for deletion from the system in the last group. However, the case with immediate end of life cycle will be moved firstly from the system before consideration of benefit value. Example, case "e" is removed from the system before other case because it is end of life case.

3.3 Two steps of maintenance process

This section illustrates the structure of Relative benefit model, which is separated into two steps to maintain case, classify case and deletion policy as illustrate in Figure 3.1

3.3.1 Classification case

In electronic shop, each case is important to sales application because it can represent product contained in actual inventory. Maintenance case base system is helps to increase the efficiency and effectiveness for sales processing especially in term of storage control, inventory control, and performance control that lead to increase customer satisfaction. To delete each case we must consider effectiveness of sales transaction and efficiency case base system. Each case should be classified in term of benefit to case base system and electronic commerce business. Some attributes and transactions of a case such as frequency of sell and frequency of retrieval, therefore, are selected to be evaluation item to model relative benefit value. There are four steps to classification case as follows

1. Select product type
2. Select factor and identify the group of quality to each factor for using in the calculation of relative benefit value
3. Define relative benefit value
4. Classification cases into several groups based on benefit level

3.3.1.1 Select product type

Sine, there should be many products sell in electronic shop, the case should be kept in case library indexed by product type. In business environment, each product may be measure either factor in different value such as high gross margin of sell for PC should not lower than 20% while high margin of printer should between 10-15%. Therefore, the maintenance case should be separately considered if possible.

3.3.1.2 Select factor to denote relative benefit value to measure the benefit of a case

The factors selected to be the criterion for measuring level of benefit to each case must contribute mainly to be benefit for the vender and the customer in term of sales processing. The criteria can be any attributes or transactions of a case depending on each business type. However, this thesis shows some example criteria to evaluate and identify the relative benefit of a case

- Case attribute such as pricing, date, time, stock, etc.
- Transaction such as frequency of retrieval, frequency of sell, etc.

Criteria / Fuzz term	**Average frequency of retrieval**	**Average frequency of sell**	**Margin**	**Life Cycle**
1st Level	Low	Low	Low	Short
2nd Level	Normal	Normal	Normal	Normal
3rd Level	High	High	High	Long

Table 3-1:The table of modeling quality to each factor

From the Table 3-1, each factor can be selected from attributes and transaction of a case occurred during sales processing. *Average of frequency of retrieval, Average of frequency of sell, margin and remaining date of life cycle* are selected to be the sets of fuzzy.

Factor 1: Average frequency of retrieval (AvgFR)

The average frequency of retrieval is a ratio of an average of frequency that the case is retrieved from the system in selling process and the period of that case available in the system.

AvgFR = total amount of retrieval/(Effective date-Expired date) (3.1)

Factor 2: Average Frequency of sell (AvgFS)

The average of frequency of sell is a ratio of an average frequency that the case is selected for buy from the buyer or customer and the period of that case available in the system

AvgFS = total amount of sell out times/(Effective date- Expired date) (3.2)

Factor 3: Margin

Margin is the percentage of margin that can earn from selling process

Margin = [(sale price - cost) x 100]/sale price (3.3)

Factor 4: Life cycle (LC)

Life cycle is the remaining day that a case is available in the system till end of its life cycle.

Life cycle = (Expired date-current date) (3.4)

After collect static of value of all factors, their value are denoted the term Low, Medium or High group of their quality level (see example in Table 3-1). The degree of membership function represents the grade of each term.

Referring to the membership function represented in (2.5), (2.6) and (2.7), they are used to calculate and denote the term Low, Medium and High for each factor.

Case no.	Code	Brand	Price	Solution	
Case01	IBMH001	IBM	4000	10GB	Hard Disk
Case02	HPH001	HP	3500	10GB	Hard Disk

Table 3-2: Case of personal computer

The element or value of each factor, the average of frequency of retrieval and frequency of sell, gross margin, and product life cycle, is collected for using in specification of quality level to such factors.

Case no	AvgFR	AvgFS	Margin	LC
Case01	0.875	0.0571	7.4074	0.4286
Case02	1.8958	0.833	10	-0.625

Table 3-3: Factors and their elements

Referring the graph of fuzzy set Low in figure 2-6, the example of defining the quality for each factor can be described in next page

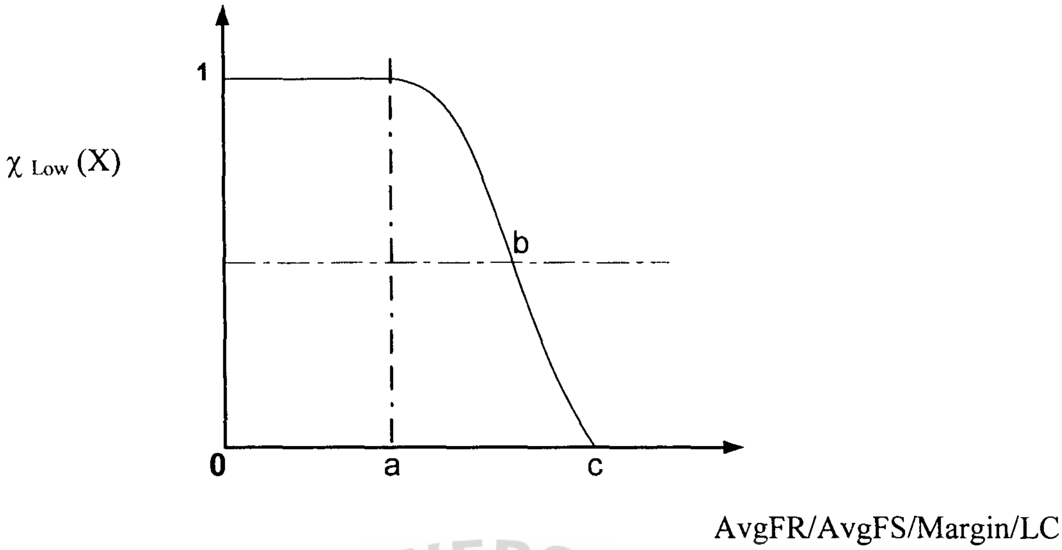


Figure 3-3: Graph of membership function for term Low of each factor

Let the range of term Low quality for each factor is set as follow:

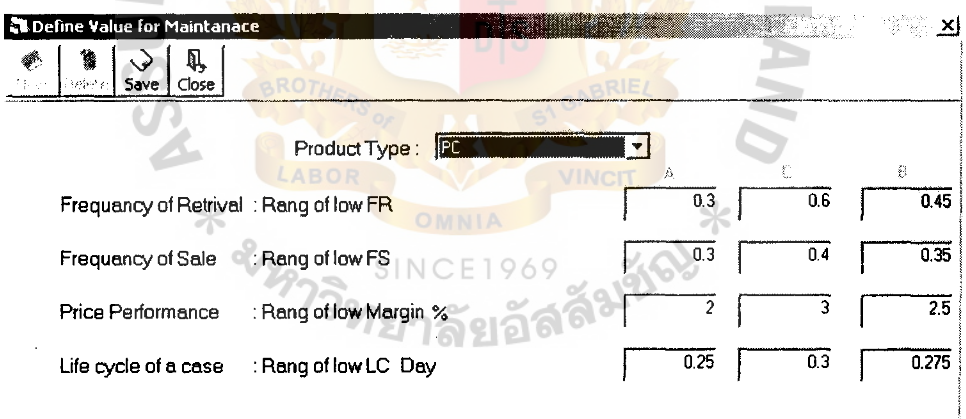


Figure 3-4: Define rang of low quality to each factor

If value of Avg FR is 0.875, we can define the membership function of fuzzy set Low by using formula (2.5)

$$\chi_A(X) \quad 1\text{-}S(0.875;a,b,c) = \begin{cases} 1 & \text{for } x \quad a \\ 1\text{-}2((0.875\text{-}a)/(c\text{-}a))^2 & \text{for } a<x<b \\ 2((0.875\text{-}c)/(c\text{-}a))2 & \text{for } b<x<c \\ 0 & \text{for } x \quad c \end{cases}$$

where "a" and "c" are the function's end points, and b = (a+c)/2

Let a = 0.3, c= 0.6, b=0.45, the degree of membership function in set low of FR value 0.875 is 0.

If we denote fuzzy term Medium and high to element 0.875 by using formula (2.10) and (2.11) in order, we can get the degree membership function of term Medium is 0 and term High is 1. Therefore, the element 0.875 should be denoted to term High because the maximum of degree of membership function belongs to term High.

3.3.1.3 The calculation of Relative Benefit Value (RB)

After the quality of each factor is denoted such as Margin is high or Frequency of retrieval is high, these results can be used to define Relative Benefit Value of each case.

Let I = { A_1, A_2, A_3,..A_n}; A_1,A_2,..A_n is set of transaction or case attribute selected as factors to measure benefit of each case.

Dgr = degree of membership of each evaluation item

W = weight of each classification quality;

where $W_{Low} < W_{Medium} < W_{High}$

N = amount of evaluation items

$$\textit{Relative benefit value } \textbf{(RB)} = \frac{\sum_{i=1}^{N}(W_i * Dgr_i)}{N} \tag{3.5}$$

The format of distribution form can be shown as in this following formula

$$RB = [(W_{A_i} * Dgr_{A_i}) + (W_{A_2} * Dgr_{A_2}) + (W_{A_3} * Dgr_{A_3}) + \ldots (W_{A_n} * Dgr_{A_m})] / N \tag{3.6}$$

Example of how to calculate relative benefit value

Let “case A” have 0.8 of a degree of membership function of LOW for frequency of retrieval, a degree of membership function of set HIGH for frequency of sell is 0.7, a degree of membership function of set HIGH for price performance is 0.9, a degree of membership function of set MEDIUM for life cycle is 0.7 for life cycle, the weight for LOW class is 1, the weight of MEDIUM class is 2 and the weight of HIGH class is 3. The relative benefit of case “A” can be defined as follow

If $I = \{ AvgFR, AvgFS, Margin, LC\}$

where Avg*FR* is average frequency of retrieval, Avg*FS* is average frequency of sell, *Magin* is margin of sales, and *LC* is remaining of Life cycle of a case

$$RB = (W_{AvgFR}*Dgr_{AvgFR}) + (W_{AvgFS}*Dgr_{AvgFS}) + (W_{margin}*Drg_{margin}) + (W_{LC}*Dgr_{LC})]N$$

$$= [(1x0.8) + (3x0.7) + (3x0.9) + (2x0.7)]/4$$

$$= 1.75$$

3.3.1.4 How to classify case

Each case can be classified in to the groups depending on relative benefit value of either case. The groups that are denoted to the level of benefit are classified by using the three linguistic words, Low benefit, Medium benefit, and High benefit, according to the subjective judgment of the inspector.

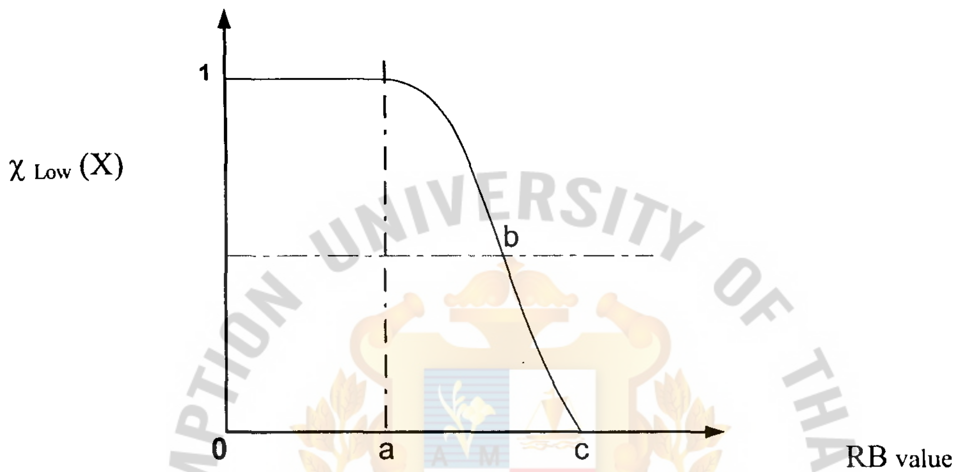


Figure 3-5: Graph of membership function of RB value in set LOW

Referring to the formula (3.10), let X be Relative benefit value. The low RB is used for guiding case deletion. The key concept is classified into the group LOW RB which containing the case with about low RB value. The degree of membership function of $\chi_{LOW}(X)$ is between 0 to 1. The fuzzy set LOW of X is the set characterized by the membership function

$$\mu Low: X \rightarrow [0,1]$$

Here, the degree of membership is a measure of benefit level, and the best quality level is 0 and the worst 1[17]; it is a basic variable standardized on the interval [0,1]. An example of membership functions for the three linguistic variables mentioned above is shown in following:

Let x be degree of membership function of Relative benefit value that variable standardized on the interval [0,1], y is membership function of degree of membership function in low benefit level, medium benefit level and high benefit level.x1 and x2 is the minimum and maximum of x that classified into Low benefit group, x3 and x4 is the minimum and maximum of x that classified into Medium benefit group, x5 and x6 is the minimum and maximum The group of Low benefit, Medium benefit and High benefit can be denoted from the degree of membership function as shown in figure 3-6

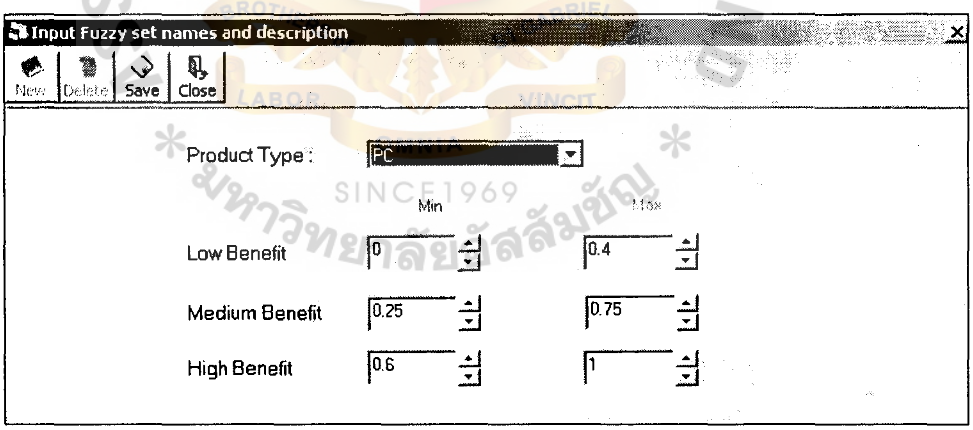


Figure 3-6: The minimum and maximum of fuzzy term Low, medium, and High benefit

Let x1 = 0, x2 = 0.4 x3 = 0.25, x4 = 0.75, x5 = 0.6, and x6 = 1, The shape of benefit level graph also can be subjects in Figure 3-6.

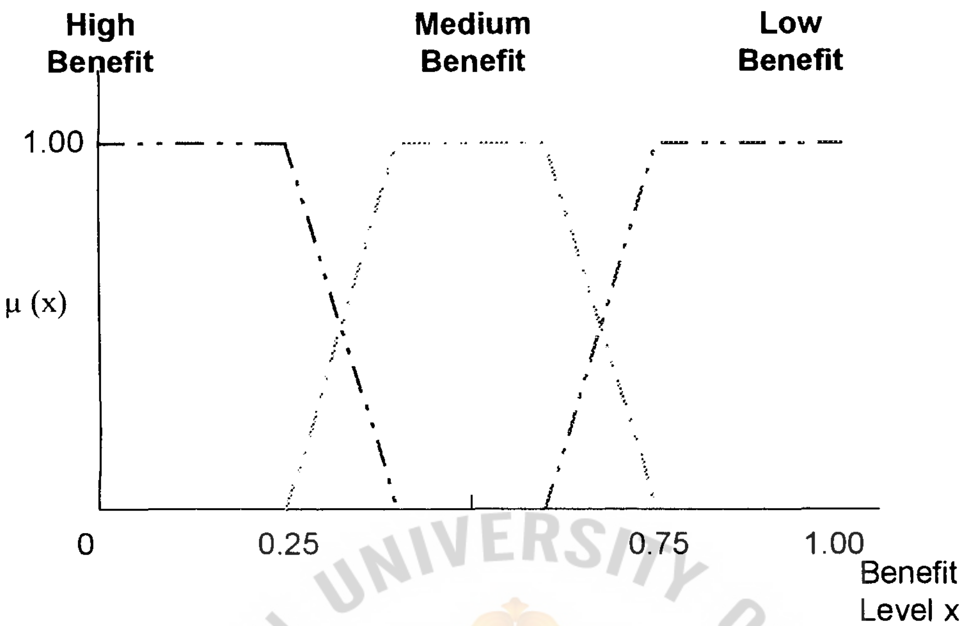


Figure 3-7: Graph of relative benefit level

Referring to fuzzy set theory [16] that operated the union of two fuzzy sets is the set of their elements that belong either to one of the constituent sets or to both. For any point in the domain of the two fuzzy sets, the membership grades of the elements of the union set must be equal to the greater of the two membership grades of the either one of the constituent sets. Thus,

The union of the fuzzy sets A and B, denoted by $A \cup B$ (or A OR B) is defined by:

$$A \cup B = \int_U [\chi_A(x) \ \# \ \chi_B(x)]/x, \qquad x \in U \qquad (3.7)$$

where the symbol # stands for max, Thus, for any $a \in A$ and $b \in B$,

$$max\,[\chi_A(a), \chi_B(b)] = \chi_A(a) \qquad \text{if } \chi_A(a) \geq \chi_B(b) \qquad (3.8)$$

$$= \chi_B(b) \qquad \text{if } \chi_A(a) < \chi_B(b) \qquad (3.9)$$

The max operation can exhibit the following property

Associativitity: $[\chi_X(x) \cup \chi_y(y)] \cup \chi_z(z) = \chi_X(x) \cup [\chi_y(y) \cup \chi_z(z)]$ (3.10)

where X, Y, and Z be fuzzy sets and let $x \in X$, $y \in Y$, and $z \in Z$

Let X, Y, and Z are fuzzy sets of Low benefit, Medium Benefit and High benefit in ordered. x, y, and z are relative benefit value of the cases where $x \in X$, $y \in Y$, and $z \in Z$. Thus, the cases are in

Low benefit group if $\chi_{Low}(x) \geq \chi_{Medium}(y)$ and $\chi_{Low}(x) \geq \chi_{High}(z)$

Medium benefit group if $\chi_{Medium}(y) \geq \chi_{Low}(x)$ and $\chi_{Medium}(y) \geq \chi_{High}(z)$

High benefit group if $\chi_{High}(z) \geq \chi_{Low}(x)$ and $\chi_{High}(z) \geq \chi_{Medium}(y)$ (3.11)

3.3.2 Deletion policy

This section is detailed in the deletion policy based on cases categorized in previous section. Case deletion is favoring cases with low RB values.

Ideally, the case in Low benefit is considered for deletion before the case in medium benefit and high benefit group. However, the case with immediate end of life cycle will be the first priority to be selected for deletion. Within each group, the case that has lowest relative benefit value will be selected before higher value in order.

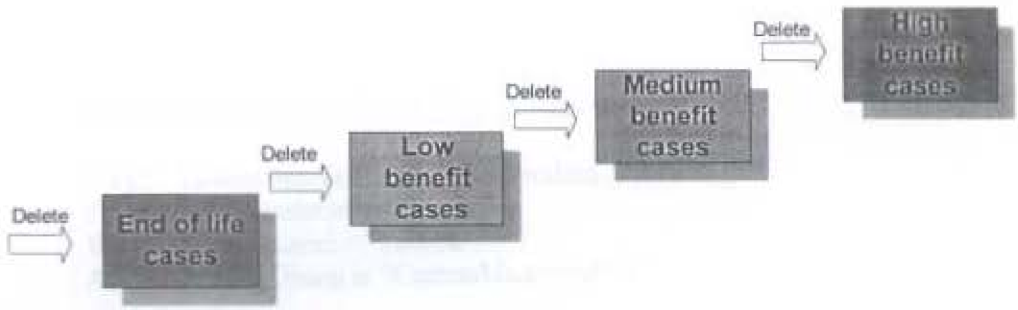


Figure 3-8: Deletion diagram

In the deletion process, the selection case or group will be moved to recycle bin area before removing from the system. The system administrator can review the cases that were deleted again. They can still use the case from recycle bin again.

Let case base represent base product (Single), package product, or custom made product customized by customer during searching in sale process, the policy to maintain either case based on relative benefit value is shown in the following procedure

Procedure DeleteCase IS

```
Begin
    If      there are EndOfLife Case        Then
            Call DeleteSingleProduct(EndOfLifeSingleProduct)
            Call DeletePackageProduct(EndOfLifePackageProduct)
            Call DeleteCustomMadeProduct(EndOfCustomMadeProduct)
    Elseif  there are  Low Benefit Group    Then

            Select  LowestBenefit.SingleProduct
            From    LowBenefitGroup Product
            Where   ProductLevel = "Lowest"
             And    ProductGroup is "SingleProduct "

            Call SingleProduct(LowestBenefitProduct)

            Select  LowestBenefit.PackageProduct
            From    LowBenefitGroup Product
            Where   ProductLavel = "Lowest"
            And     ProductGroup is "PackageProduct "
```

Call PackageProduct(LowestBenefitProduct)

```
Select  LowestBenefit.CustomMadeProduct
From    LowBenefitGroup Product
Where   ProductLavel = "Lowest"
And     ProductGroup is "CustomMadeProduct "
```

Call CustomMadeProduct(LowestBenefitProduct)

```
Elseif  there are  Medium Benefit Group          Then

        Select  LowestBenefit.SingleProduct
        From    MediumBenefitGroup Product
        Where   ProductLavel = "Lowest"
        And     ProductGroup is "SingleProduct "
```

Call SingleProduct(LowestBenefitProduct)

```
Select  LowestBenefit.PackageProduct
From    MediumBenefitGroup Product
Where   ProductLavel = "Lowest"
And     ProductGroup is "PackageProduct "
```

Call PackageProduct(LowestBenefitProduct)

```
Select  LowestBenefit.CustomMadeProduct
From    MediumBenefitGroup Product
Where   ProductLavel = "Lowest"
And     ProductGroup is "CustomMadeProduct "
```

Call CustomMadeProduct(LowestBenefitProduct)

```
Elseif  there are  High Benefit Group    Then

        Select  LowestBenefit.SingleProduct
        From    MediumBenefitGroup Product
        Where   ProductLavel = "Lowest"
        And     ProductGroup is "SingleProduct "
```

Call SingleProduct(LowestBenefitProduct)

```
Select  LowestBenefit.PackageProduct
From    MediumBenefitGroup Product
```

Where ProductLavel = "Lowest"
And ProductGroup is "PackageProduct "

Call PackageProduct(LowestBenefitProduct)

Select LowestBenefit.CustomMadeProduct
From MediumBenefitGroup Product
Where ProductLavel = "Lowest"
And ProductGroup is "CustomMadeProduct "

*Call CustomMadeProduct(LowestBenefitProduct)*End if;
End if;
End if;

End;
End DeleteCase

Procedure Deleted SingleProduct (P_ProductCode)IS

<Define Cursor><Base product Cur>
Select EndofLife.Product
From AllProduct
Where ProductGroup is "SingleProduct " AND ProductCode = "P_ProductCode"

<Open Cursor read data>< Single product Cur >

Execute Command Fetch data
Fetch data from Single product Cur into variable as following declaration part
Execute Command Delete
Loop till either send status is success or Cursor % not found
Execute command do commit data
Terminate with Success.
End Deleted SingleProduct

Procedure Deleted PackageProduct (P_ProductCode)IS

<Define Cursor><Package product Cur>
Select EndofLife.Product
From EndOfLifeProduct
Where ProductGroup is "PackageProduct "

<Open Cursor read data><Package product Cur>

Execute Command Fetch data
Fetch data from into variable as following declaration part
Execute Command Insert into All Delete Code
Insert into COLUMNS

Product_code,
Product_Description,
Delete Date

Execute Command Delete

Loop till either send status is success or Cursor % not found
Execute command do commit data
Terminate with Success.

End Deleted PackageProduct

Procedure Deleted CustomMadeProduct (P_ProductCode)IS

<Define Cursor><CustomMade product Cur>
Select EndofLife.Product
From EndOfLifeProduct
Where ProductGroup is "CustomMadeProduct "

<Open Cursor read data><Package product Cur>

Execute Command Fetch data
Fetch data from into variable as following declaration part
Execute Command Insert into All Delete Code
Insert into COLUMNS
Product_code,
Product_Description,
Delete Date

Execute Command Delete

Loop till either send status is success or Cursor % not found
Execute command do commit data
Terminate with Success.

End Deleted CustomMadeProduct

Procedure Alert Deleted Product(P_ProductCode Parameter In) IS

<Define Cursor><Alert product Cur>
Select ProductCode
From All Delete Product {Summary of all product}
Where ProductCode = P_ProductCode

Popup Alert message ('Product was delete is' || Product Code || Product Description)

Exception When no data found Then
Null;/* Product existing in system */

<Open Cursor read data><AlertProduct Cur>

Execute Command Fetch data
Fetch data from into variable as following declaration part
Execute Command Delete

Loop till either send status is success or Cursor % not found
Execute command do commit data
Terminate with Success.

End Alert Deleted Product

Figure 3-9: The algorithm of deletion each case

3.3.3 Relative benefit in adding case into new case library

This method is reverting of deletion case. The cases in an original case base are selected and added to an empty case base until the limited size is reached. Selection can be one case or whole group. The high benefit case will be the first group for selection, and then followed by medium benefit case and low benefit case in order. The detail is not described in this thesis.

CHAPTER 4: EXPERIMENTAL AND EVALUATION

As the performance evaluation on this type of contribution is difficult because in domain of electronic shop the performance of retrieval case is involved with hardware specification of remote client. This chapter will focus on measuring the quality of remaining cases after deletion case from the system based on beneficial value used as a criterion. For evaluation, the relative benefit method is compared with the utility method on different sizes of case library.

4.1 Environment of Examination

Regarding the measurement of evaluation that focused on the quality of remaining case after deletion, the following issues are the consideration for evaluation

- The expected results of cases after deleting target case from case library
- The deletion are examined with different amount of fixed sizes of cases library (swamping limit)

The environment used to evaluate the measurement of the quality of remaining case and response time is examined under the following system environment:

Hardware specification

- Intel Pentium II 600
- Hard disk 12 GB
- RAM 128 MB

Software specification

- Using Microsoft Visual Basic 6.0 develops the prototype.
- A tested database in on Microsoft SQL server
- Windows 2000 Professional

Tested case base system

The case library system is stimulated to initial case. A case is initialed from both original server and via web page. The database is developed based on specific and control domain of PC/Server configuration. Each case constrains necessary attributes of PC/Server specification. Case name, brand, type, product number, description, CPU, memory, monitor, modem, hard disk, price, effective date, expired date, etc, are all attributes within a case. An average of frequency of retrial and frequency of sell per day, the margin of sales, and remaining day that each case stay in the system were kept a static as a factor to measure a beneficial case for sales processing in electronic commerce.

4.2 Experimental method

An initial case-base size of 25 cases was creased. The web accessing for searching and doing sales transaction is started. Additional cases are added to the system until the size of case base hits 50 cases, swamping limit. The example of description of a case is as follow :

Case Name	Brand	Description	Monitor	Modem	Ram	Harddisk	CPU	CD-ROM	DVD-Rom	CD-RW	LAN	Price
P0001	Compaq	Pentium III no	15" Compa	56 K Comp	128 Comp	20 Compa	133 MHz	-	-	-	-	31200
P0002	Compaq	Pentium III CD	15" Compa	56 K Comp	128 Comp	20 Compa	133 MHz	48x Comp	-	-	-	32200
P0003	Compaq	Pentium 4 1.6G	15" Compa	56 K Comp	128 Comp	20 Compa	1.6GHz	-	-	-	-	38000
P00031	Compaq	Pentium 4 1.6G	15" Compa	56 K Comp	128 Comp	20 Compa	1.6GHz	-	-	Yamaha	-	42900
P0007	IBM	Pentium III 800	15" IBM	56 K IBM	64 IBM	20 IBM	800 MHz	-	-	-	-	27500
P00071	IBM	Pentium III 800	15" IBM	56 K IBM	64 IBM	20 IBM	800 MHz	-	16x HP	-	-	32300
P0010	HP	Pentium III 1.3	15" HP	56 K HP	64 HP	40 Seagate	1.3 GHz	-	-	-	-	29000
P00101	HP	Pentium III 1.3	15" HP	56 K HP	64 HP	40 Seagate	1.3 GHz	-	-	-	LAN typ	35000
P00102	HP	Pentium III 1.3	15" HP	56 K HP	64 HP	40 Seagate	1.3 GHz	-	-	-	10/100 LA	36000
P0039	StarPC	Celeron 1.3Ghz	-	56 K US Ro	128 Hitach	40 Seagate	1.3 GHz	-	-	-	-	13000
P00391	StarPC	Celeron 1.3Ghz	-	56 K US Ro	128 Hitach	40 Seagate	1.3 GHz	16x Asus	-	-	-	15000
P003910	StarPC	Celeron 1.3Ghz	-	56 K US Ro	128 Hitach	40 Seagate	1.3 GHz	-	-	LG		16450
P0049	InnoPC	ATHLON 1.4 G	-	56 K Rockw	128 Hitach	20 Seagate	1.4GHz	52x Sony	-	-	-	14000
P0050	InnoPC	ATHLON 950	-	56 K Rockw	128 Hitach	20 Seagate	950 MHz	52x Sony	-	-	-	13000
P00501	InnoPC	ATHLON 950	-	56 K Rockw	128 Hitach	20 Seagate	950 MHz	52x Sony	16x HP			17800

Table 4-1: Example of case description

The cases are classified into low benefit, medium benefit and high benefit group based on relative benefit value identified from necessary factors such as average frequency of retrieval, average frequency of sales, margin, and remaining day that case available in the system. The identification of values for low quality of each factor used to define relative benefit value is as follow :

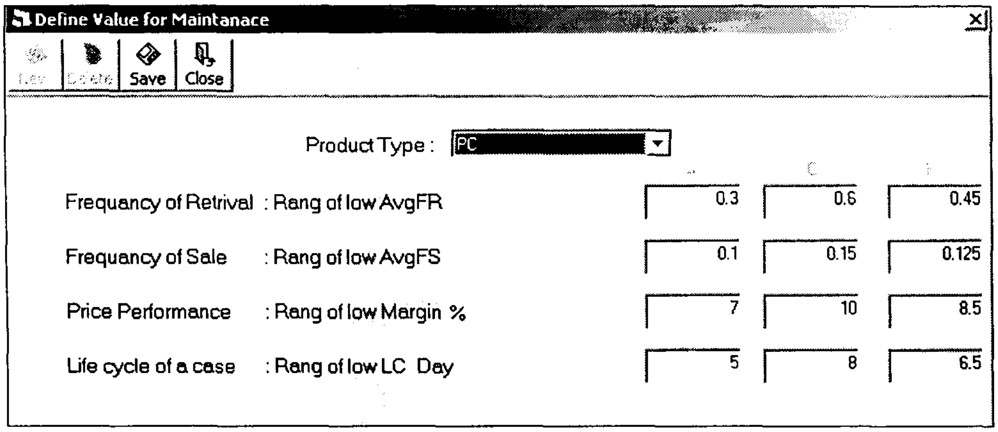


Figure 4-1: Define range of low quality to each factor

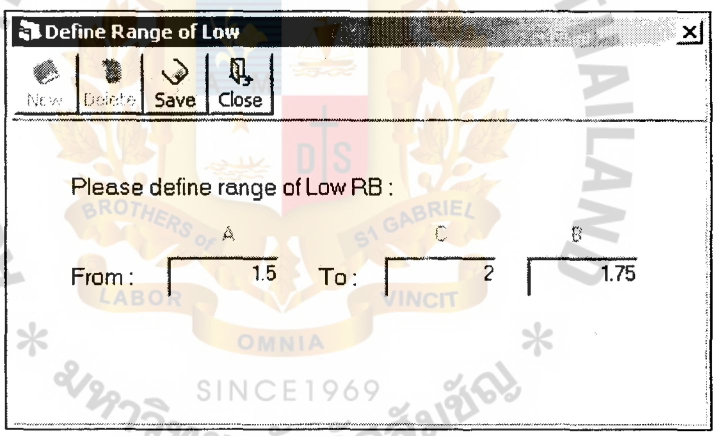


Figure 4-2: Define range of low relative benefit value

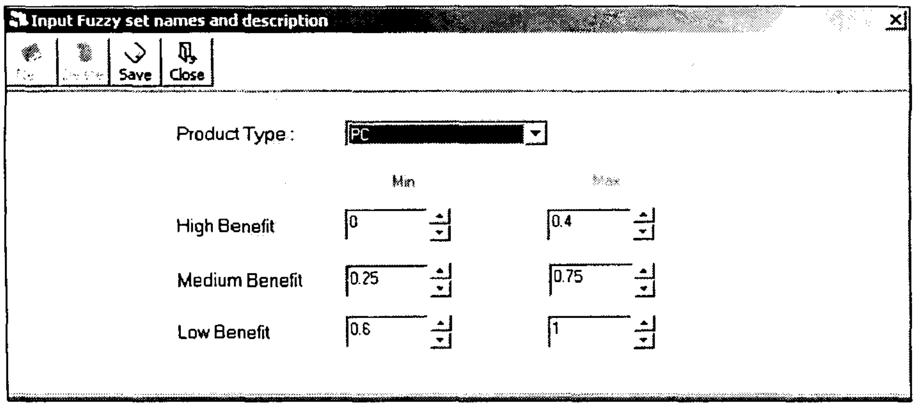


Figure 4-3: Define fuzzy set for classification of a case

The relative benefit method and the utility method are studied and experimented. The cases that remained in the system should be high benefit cases whereas deletion cases should minimize the loss of high beneficial cases. For this experiment, the cases that have top five levels of high score of relative benefit value are assumption as high benefit cases that are the criterion to measure the result of both deletion methods.

Item	Case Name	Case description	Level of benefit	Grade or degree of each level
1	P0003	PC Pentium 4 1.6Ghz	High benefit	2.6787
2	P0039	PC Celeron 1.3Ghz SAVING	High benefit	2.5
3	P0022	PC Duron 1.2 GHz	High benefit	2.5
4	P0019	PC Duron 950 MHz	High benefit	2.5
5	P0018	PC Duron 900 Mhz	High benefit	2.5
6	P0015	PC Pentium III 1.3 GHz	High benefit	2.5
7	P0008	PC Pentium III 866MHz	High benefit	2.5
8	P00071	PC Pentium III 800 MHz	High benefit	2.5
9	P00091	PC Pentium III 933 MHz	High benefit	2.4997
10	P00093	PC Pentium III 933 MHz	High benefit	2.4955
11	P0024	PC ATHLON 1.4 GHz	High benefit	2.4812

Table 4-2: Example of top five levels of high beneficial cases

The experiment starts with taking one necessary factor as a criterion to subject beneficial cases. An average frequency of retrieval (AvgFR) is the first factor selected to measure a benefit of a case. The case that has much average frequency of retrieval should be of more benefit than other cases. After deletion with relative benefit method and utility method, see and compare how many high benefit cases subjected as example in Table 4-2 are deleted or remained in the system.

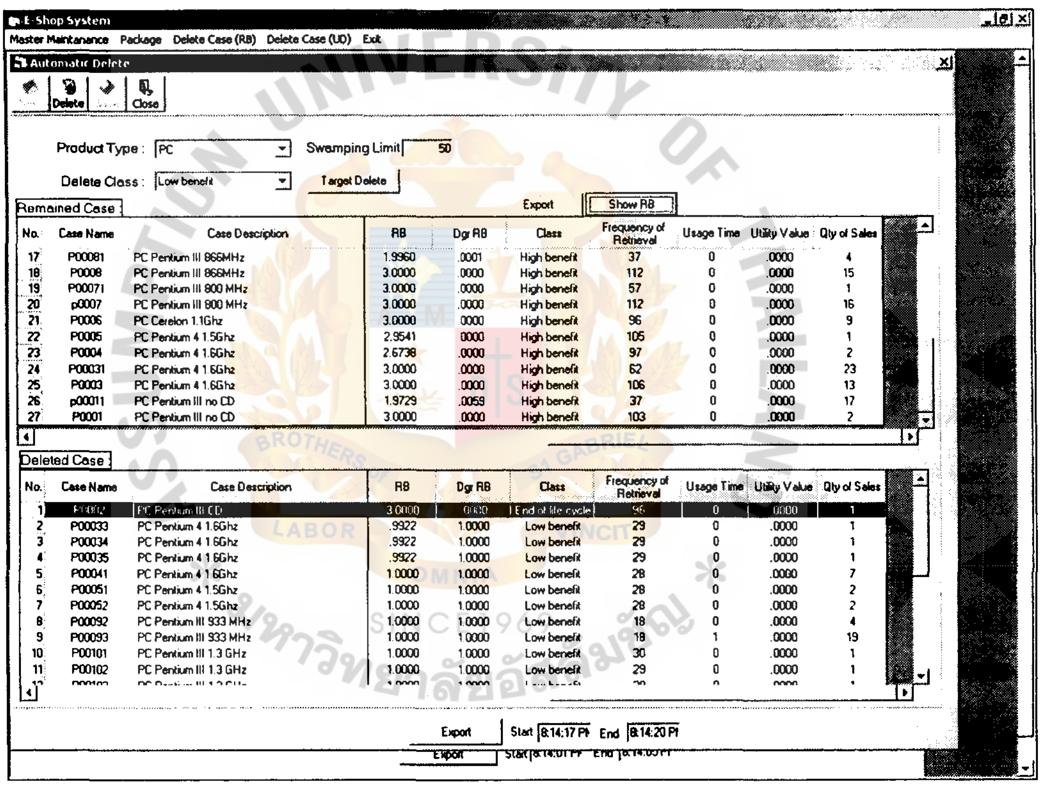


Figure 4-4: Deletion case by Relative benefit method

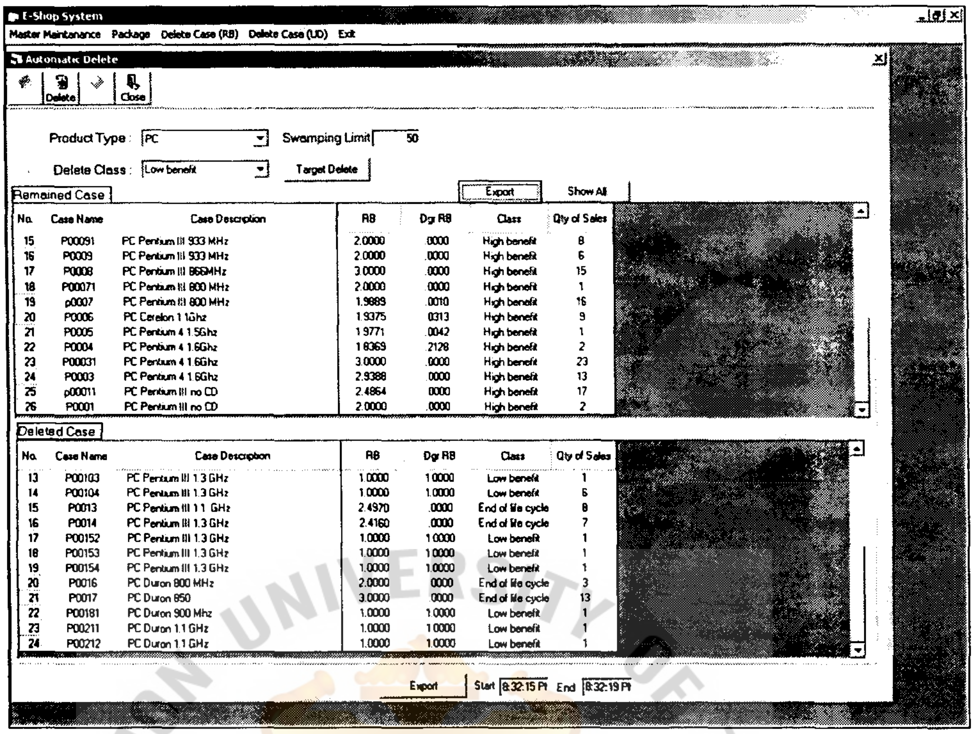


Figure 4-5: Deletion case by Utility method

Re-run the experimental again but increase the factors to be two, three, and four factors by using an average frequency of sale (AvgFS), margin, and remaining day that case available in the system (LC) to be additional factors in order.

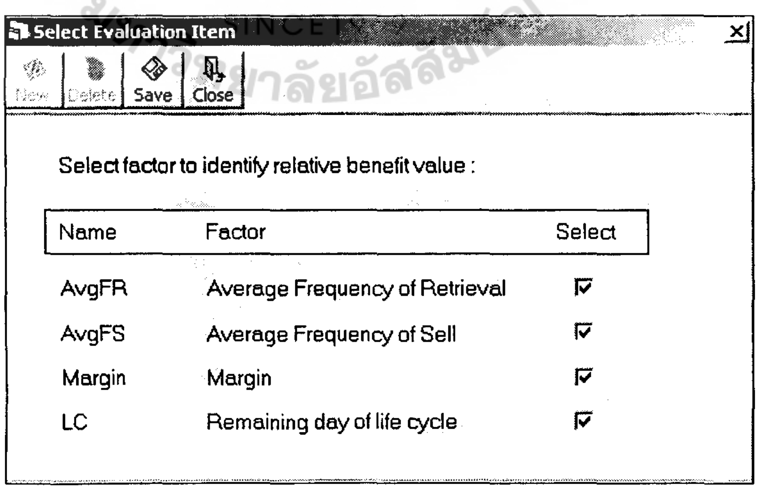
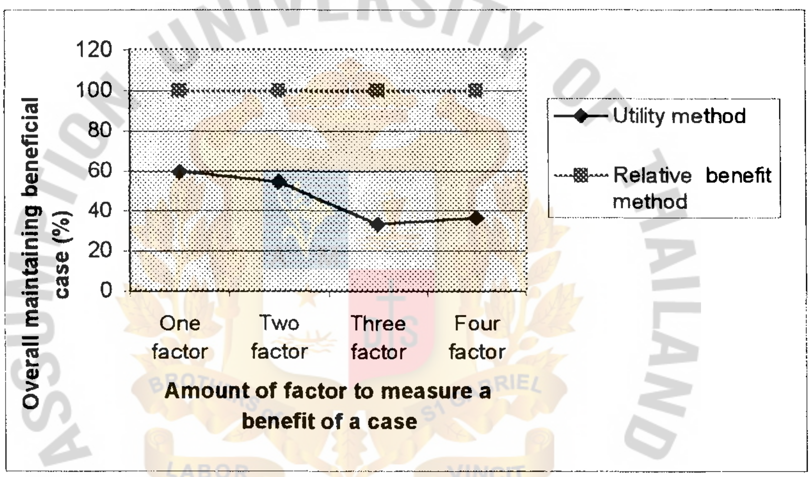


Figure 4-6: The factors selected as criteria to subject the beneficial case

Factor for measuring benefit	High beneficial case (amount)	Actual result after deletion							
		Maintaining high beneficial cases				Loss of high beneficial case			
		Utility method		Relative benefit		Utility method		Relative benefit	
		Amount	%	Amount	%	Amount	%	Amount	%
One factor	25	15	60	25	100	10	40	0	0
Two factor	22	12	54.55	22	100	10	45.45	0	0
Three factor	12	2	33.34	12	100	8	66.66	0	0
Four factor	11	3	36.37	11	100	7	63.63	0	0

Table 4-3: The result of deletion with Relative benefit method VS Utility method



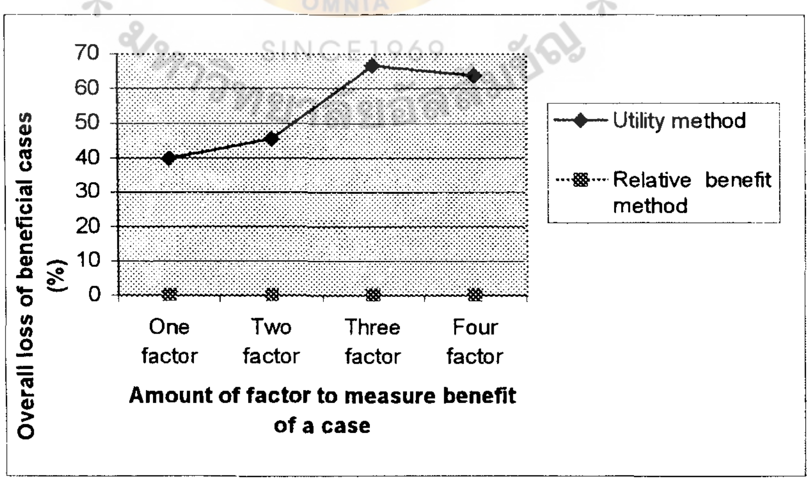


Figure 4-7: The comparison of using different amount of factors as a criterion to subject beneficial case

Experimental 2:

The previous experimental is re-studied again but both deletion methods are compared in the environment of different case base sizes, 25, 50, 75, and 100. The procedure of step for testing is as follows:

Step of Experimental

```
Get parameter;
/*from user input from terminal : Start with AvgFR till CL factor*/

Procedure ExperimentalStep (AvgFR,AvgFS,Margin,LC In,CompareResult Out) As

Begin
 /* NumberOfCase start with 25 and increase by 25 till 100 */
NumberOfCase := 25;
For    NumberOfCase.value  = 25 to 100
Loop  NumberOfCase;

       Insert case.data
       Into   case table as equal to NumberOfCase.value;

       /* Factor.values  start with 1 and increase by 1 till 4 */
       Factor.values := 1;
       For Factor.values = 1 to 4
       Loop Factor
              If      Factor.values  =      1       Then
                      FactorParameter := 'AvgFR' ;
              Elsif   Factor.values  =      2       Then
                      FactorParameter :=  'AvgFR' and 'AvgFS';
              Elsif   Factor.values  =      3       Then
                      FactorParameter :=  'AvgFR' and 'AvgFS'  and 'Margin';
              Elsif   Factor.values  =      4       Then
                      FactorParameter :=  'AvgFR' and 'AvgFS'  and 'Margin'  and 'LC';
              End if;
 /* this area will be combine all parameter that  choose by user together */
 /* Compute classification will used the  FactorParameter for calculate

 EndofLife,low,medium,high values)  */
```

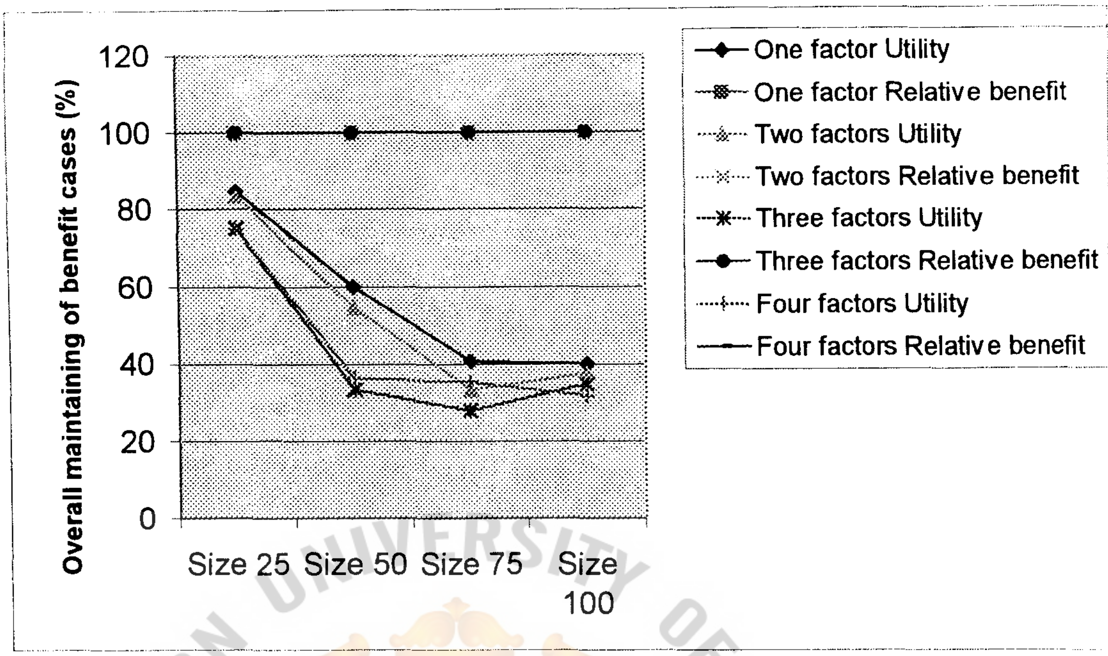
```
            Call Compute classification case(EndofLife,Low,Medium,High);
            /* Call Delete case by Relative Benefit Method */
            Call MaintainanceRB(AvgFR,AvgFS,Margin,LC);
            /*  Call Delete case by utility Method;*/
            Call MaintainanceUB Method;
            CompareResult;
            Return CompareResult;
            Increase Factor.value by 1;
            Do Factor.values till Factor.values is False ;
        Exit Factor Process;
        End Loop Factor;

 Increase NumberOfCase.values by 25;
Do NumberOfCase.values till NumberOfCase.values is False;
Exit NumberOfCase Process;
End Loop NumberOfCase;
End ExperimentalStep Procedure ;
```

Figure 4.7: Algorithm of ExperimentalStep

	Factor for measuring benefit	High beneficial case (amount)	Actual result after deletion							
			Maintaining high beneficial cases				Loss of high benefit case			
			Utility method		Relative benefit		Utility method		Relative benefit	
			Amount	%	Amount	%	Amount	%	Amount	%
Size 25	One factor	13	11	84.62	13	100	2	15.38	0	0
	Two factor	6	5	83.34	6	100	1	16.66	0	0
	Three factor	8	6	75	8	100	2	25	0	0
	Four factor	8	6	75	8	100	2	25	0	0
Size 50	One factor	25	15	60	25	100	10	40	0	0
	Two factor	22	12	54.55	22	100	10	45.45	0	0
	Three factor	12	2	33.34	12	100	8	66.66	0	0
	Four factor	11	3	36.37	11	100	7	63.63	0	0
Size 75	One factor	32	13	40.63	13	100	19	59.37	0	0
	Two factor	30	10	33.34	10	100	20	66.66	0	0
	Three factor	18	5	27.78	5	100	13	72.22	0	0
	Four factor	17	6	35.3	6	100	11	64.7	0	0
Size 100	One factor	35	14	40	35	100	21	60	0	0
	Two factor	32	12	37.5	32	100	20	62.5	0	0
	Three factor	23	8	34.79	23	100	15	65.21	0	0
	Four factor	22	7	31.82	22	100	15	68.18	0	0

Table 4-4: The result of deletion with Relative benefit method VS Utility method in different case base sized



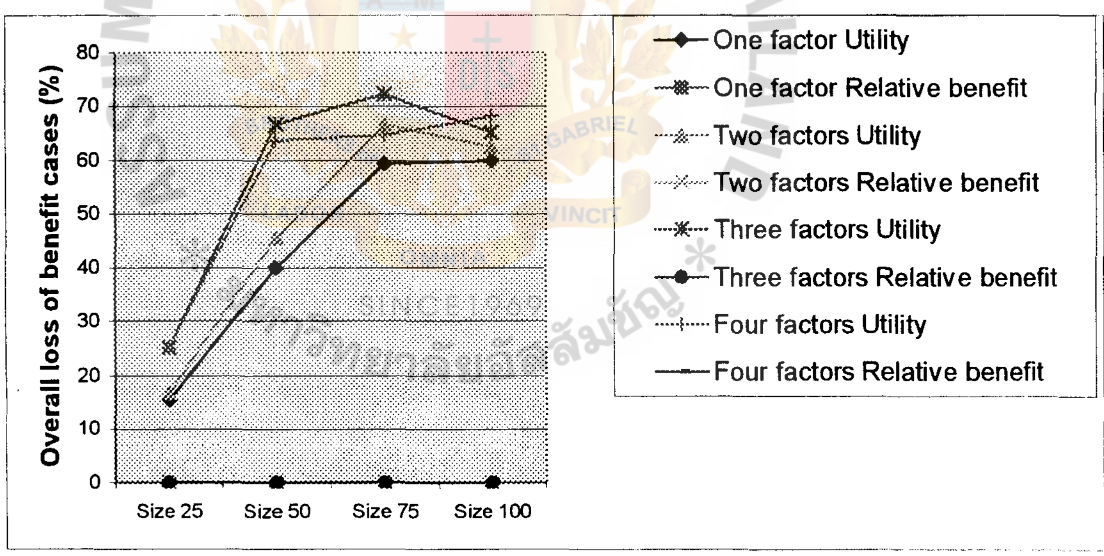


Figure 4-9: The comparison between Relative benefit method and Utility method in different sizes of case base

4.3 Result Analysis

From the previous experimental, the result can be discussed as below:

Clearly, the relative benefit based method is maintaining the cases that have high benefit in a viewpoint of sales for the domain of electronic shop whereas utility method maintain the beneficial cases less than relative benefit method because it has no understanding of the benefit in a viewpoint of sales, therefore, the high beneficial cases may be deleted from the case base system.

Figure 4-7 illustrates the graphs of comparison between deletion by relative benefit method and utility method in different amount of factors used as criterion to subject beneficial cases. The relative benefit method still can maintain the high beneficial cases. In contrast, some high beneficial cases can be deleted from the system by the utility method.

Figure 4-9 shows that the relative benefit still works very well in the incremental of case bases size. The cases deleted by this method are firstly selected and measured that they are low beneficial cases concerning with various factors. The high beneficial case must not be deleted from the system whereas the high beneficial cases may or may not be deleted from the system by utility method.

CHAPTER 5: CONCLUSION AND FUTURE WORK

In case base maintenance, there are many current researches focusing on the competence of case base system and its performance. This thesis started with an argument that the competence and performance of a case is not mainly contribution in every domain especially in sales processing within electronic shop environment. In fact, the performance of retrieval is deepened on the speed of Internet linked and system performance of customer side. This paper shows that case attributes and transactions of a case can be a practical use to identify relative benefit value for guidance either case in deletion policy. This thesis also described how fuzzy theory is valued to subject and classify the benefit level to each case in incremental updated of case base.

The relative benefit based method focuses on taking necessary factors involved in sales processing in electronic shop such as average frequency of retrieval, average frequency of sell, margin, and stilling sold to be a criterion for using to identify the relative benefit value used to measure and subject beneficial case.

The high beneficial cases must be maintained in the system whereas deletion should minimize the loss of high beneficial cases. The case that has low relative benefit value or low benefit group is selected for deletion. However, the study of stopping the intrusive reduction is important advances for case base maintenance in the future study.

BIBLIOGRAPHY

1. Smyth, B. (1998) Case-Base Maintenance. Proceedings of the 11th International Conference on Industrial & Engineering Applications of Artificial Intelligence & Expert Systems. Castellon, Spain

2. Schmitt, S. and Bergmann, R. (1999) Applying Case-Based Reasoning Technology for Product Selection and Customization in Electronic Commerce Environments. Global Network Organizations, Twelfth International Bled Electronic Commerce Conference. Bled, Slovenia, June 7-9.

3. Stahl, A., Bergmann, R., and Schmitt, S. (2000) A Customization Approach for Structured Products in Electronic Shops. Electronic Commerce: The End of the Beginning. 13th International Bled Electronic Commerce Conference. Bled, Slovenia, June 19-21.

4. Kolodner, J. (1993) Case-Based Reasonning. ISBN 1-55860-237-2

5. Riesbeck, C. and Schank, R. (1989) Inside Case-Based Reasoning.Hillsdale, N.J.: Lawrence Erlbaum Associates, Cambridge, MA

6. I.Iglezakit, I. and Anderson, Christina E. (2000) Towards the use of case properties for maintaining case-base reasoning system. Proceedings of the 6th Pacific Knowledge Acquisition Workshop (PKAW2k), pp. 135-146

7. Mckenna, E and Smyth, B. (1998) Modeling the Competence of Case-Bases. Proceedings of the 4th European Workshop on Case-Based Reasoning, Springer-Verlag.

8. Racing, K. and Yang, Q. (1997) Maintaining Unstructured Case Bases. Proceedings of the Second International Conference on Case-Based Reasoning (pp. 553--564). Providence, RI: Springer.

9. Smyth, B. and Keane, M (1995) Remembering to forget: A competence-preserving case deletion policy for case-based reasoning systems. In Proceedings of the Thirteenth International Joint Conference on Arti#cial Intelligence, pages 377#382, San Francisco, August 1995. Morgan Kaufmann.

10. Yang, O. and Zhu, J. (1999) Remembering to add: Competence--preserving case addition policies for case base maintenance. In Proceedings of the International Joint Conference in Artificial Intelligence (IJCAI), 1999. 4.

11. S. Minton. (1990) Qualitative Results Concerning the Utility of Explanation-Base Learning. Artificial Intelligence. 42, pp. 363-391

12. Wike, W Bergmann and Wess, R. (1998) Negotiation During Intelligent Sales Support with Case-Based Reasoning. Proceedings of the 6th German Workshop on CaseBased Reasoning, GWCBR'98, Germany.

13. Leake, David B and Wilson, David C. (2001) Remembering Why to Remember: Performance-Guided Case-Base Maintenance. Advances in Case-Based Reasoning: Proceedings of EWCBR-2K, Springer-Verlag, Berlin, 2000

14. Brighton, H. and Mellish, C. (2000) Identifying competence-critical instances for instance-based learners. The University of Edinburgh, UK.

15. F.Martin and Mcneill.Ellen Thro. Fuzzy Logic A Practical Approach

16. Schneider, M., Kandel, A., Langholz, G., and Chew, G. (1996) Fuzzy Expert System Tools. ISBN 0-471-95867-0

17. Asai, K.(1995) Fuzzy Systems for Management. ISBN 4 274 90033 9 C3000 (Ohmsha)

18. Aminzadeh, F. and Jamshidi, M. (1994) Soft computing Fuzzy Logic, Neural Networks, and Distributed Artificial Intelligence. ISBN 0-13-146234-2

19. Strauss, J. and Frost, R. (1999) Marketing on the Internet Principle of Online Marketing. ISBN0-13-010585-6

20. Cohan, Perter S. (2000) E-Profit High Payoff Strategies for Capturing the E-Commerce Edge. ISBN0-8144-0544-4