ABSTRACT

Artificial Intelligent is relatively new field in computer science in our country although it has been established for years in other countries. The successful implementation of artificial intelligent methods in a number of domains, especially in advisory domain, has inspired the creation of a Web-based advisory system based on artificial intelligent methods, hence, Course Advisory System (CAS).

The main purpose of CAS is to enable STPM students that are applying for courses during the university in-take to get advice through the use of Internet thus given students a better chance of securing a place in university. The system also enables the user to search for synopsis on the courses.

ACKNOWLEDGEMENT

Many people contributed their precious time, gave freely of their invaluable advice and, in one manner or another, helped me towards the completion of this report. So, by this opportunity, I would like to express my gratitude and thanks to the following persons.

Firstly, I would like to express my deepest gratitude to respected supervisor Puan Norisma Idris for her invaluable guidance and advices in completing this report. Then I also would like to address thankfulness to my appreciated moderator, DR. Rukaini Haji Abdullah for her invaluable comments and advices.

Then, I would like to thanks to who had consumed their invaluable time to share their information and experience with me during the interviews section held. Here, I sincerely say thanks to Miss Then San San, member of the educational counselling organization.

Next, I like to deliver thanks to all my friends and also my course mates like Miss Kong Sing Teang, Miss Kong Sing Eming, Mr. Khor Shen Shin, Mr. Loo Hoi Keong, Miss Lee Suat Fen for their encouragement and also their helping hands whenever I faced problems.

Finally, I would like to express my highest appreciation to my family members for their patient, consideration, support and encouragement in order to enable me to complete this report.

TABLE ON CONTENTS

Chapter 1 Introduction

Chapter 2 Literature Review

Chapter 3 Methodology

Chapter 4 System Analysis

LIST OF FIGURES

LIST OF TABLES

Chapter 1: Introduction

1.1 Introduction

University Malaya (UM) is the most famous and eldest university in Malaysia. Each year, a lot of students, especially those who have passed their Sijil Tinggi Pelajaran Malaysia (STPM), would apply for courses offered by UM. Problem arises when there is not proper guidance given to the students where students will blindly apply for courses offered.

Manual guideline given by the Bahagian Pengurusan Kemasukan Pelajar IPT is also proven inadequate to help students in choosing suitable courses. With these problems in mind, the author has purpose to develop the Courses Advisory System (CAS). CAS is intended to help students in making decision regarding the courses that they should apply by giving advice base on previous experience.

1.2 Problem Definition

***Academic Advisement* has played an important role in students' lives.** This statement has been especially true for those students who have passed their Sijil Tinggi Pelajaran Malaysia (STPM) and are interested in enrolling into local universities to further their studies. They are usually in dilemma of choosing the most suitable course for themselves. Choosing a suitable course is very essential to avoid regrets later in their life. However, without proper guidance and advice, students tend to make the wrong kind of decisions, only to realize the mistake after enrolling. By this time, it is usually too late to make any changes.

Therefore, to avoid making mistakes, students will usually try to get some advice from seniors who are either studying or have been studying in the university. Some will try to get some insight of the courses that their seniors are pursuing in the university. Since undergraduate normally don't have an overview of all the other courses that are offered in their university, their advice is limited to their own courses respectively. Besides, the advice given is only based on their own experiences and thoughts hence, making their advice unreliable.

There are also students who would seek counselling from their schools counselling teachers. With help from their teachers, most students may have a better understanding of the courses that are offered by the university. However, this does not provide students with a more statistically accurate of chances of enrolling in those courses.

Another source of information that can be considered reliable is the Educational Counselling Organizations. This has also been my source of inspiration in conducting this project. Throughout the years, these organizations have been conducting surveys and analysing data to provide a more accurate view of chances for students to enrol in to a particular course of their choice. The problem faced by these organizations is lack of a good communication method to convey this information to students. A normal method that these organizations used is the call-in method, where students will have to call in to these organizations. This method is lack of efficiency because the operators from the organizations will have to understand all the courses manually i.e. by reading and asking. Since students are calling from all over Malaysia, this method has proved to be very costly for them. Most of them become hesitant once they think about the cost. They would prefer to gain information from other sources.

1.3 Project Aim and Objective

With this project, I hope that it can be useful in helping students in applying courses in the University of Malaya. The aims of this project can be divided into 3 main groups: to those students that are going to apply for courses in University of Malaya, to the university, and also to the author.

1) **The aim of this project for the benefits of students**

i) To help students in applying the most suitable courses for themselves in term of their interests and also their results. With this, hopefully students will have bigger chances of entering the university.

ii) To provide students with an overview of the courses offered.

2) **The aim of the project for the benefits of university**

i) As a way for the university to give a better overview of the courses offered in the university.

ii) To promote the courses which offered by University Malaya.

3) **The aim of the project for the author benefits**

i) To help the author in further understanding of the use of Artificial Intelligent technique namely the Case Based Reasoning technique.

ii) As a way for the author to learn the practical used of Case Based Reasoning after learning the theory behind CBR.

iii) To have a first hand experience on software development as the author has to cover every aspect of the software development life cycle. It also gives opportunity to utilize what the author has learned during the System Analysis and Development course and also the Software Engineering course.

iv) To learn about the language that can be used to develop web based application and also the used of shells.

1.4 Project Scope

The scopes of the project are as follows:

- The system that is going to be developed is aimed to provide advice to students who are going to apply for courses in the university, specifically students that have passed the Sijil Tinggi Pelajaran Malaysia (STPM). Students from STPM level who are interested to apply for courses that are offered by University of Malaya can use the system to get information of the courses that they are most likely to get offered if they apply for them. This can help to increase their chances in getting into University of Malaya.
- The system is developed as a web-based system so that it can be accessed by all the students throughout Malaysia.
- The courses that provided by the system are only those offered by University of Malaya.
- The system can be used by both teachers and parents as a guide by explaining the courses that are offered in the University of Malaya. It can be used also by those students that are going to sit for their STPM. They can know the minimum requirement to enter the courses that they interested in.

- The system owners can be the administration of the University of Malaya where they can use the system as a way of providing information of the courses in the university to the public. Besides that, with the use of the system, students that are going to enrol into the courses are likely to be more inform regarding the courses which can reduce cases where students apply for changing of courses after they enrolled in to the university.
- Besides that, other target owners are those educational counselling organizations which throughout the years have provided students with information in the courses of the university.

1.5 Project Assumption

There are certain assumptions that are made about the users and administrator of this system.

The assumptions that are made about the users are as follow:

i) Students that have passed and obtained their Sijil Tinggi Pelajaran Malaysia (STPM) result.

ii) Students that are interested in enrolling in to University of Malaya.

Assumptions that are made about the administrator of the systems are as follow:

iii) The administrators have a certain level of knowledge of using the programming language that is going to be used.

iv) The administrators have a certain level of knowledge on the used of Case Based Reasoning (CBR).

v) The administrators have conducted survey on the students to obtain the cases that are needed in the CBR.

1.6 Motivation Factors

The motivation and inspiration that have given me the idea of developing this system are:

i) To explore and learn the used of Case Based Reasoning especially to be used to solve real life problem.

ii) To learn the used of programming language in developing a website and also to used the language to implement the used of CBR.

iii) To help students in getting into courses that they like and also to help students getting a bigger chance of entering University of Malaya.

1.7 Report Layout

This report has been divided into 9 chapters which are introduction, literature review, methodology, system analysis, system design, system development, system testing, discussion and finally appendix.

The first chapter, **Introduction**, give a definition on the problems, aim or objective of the project, project scope and also the outline of the project development.

Chapter 2, **Literature Review**, talks about problems analysis before the project is underway. It includes research and analysis on previous system, research on the methods that will be used and also research on the domain of the problems.

Chapter 3, **Methodology**, is an in depth elaboration on the research methods and technique that is used in the project.

Chapter 4, **System Analysis**, is about the requirement of the system which includes functional requirements, non-functional requirements and also the hardware and software requirement of the projects.

Chapter 5, **System Design**, discuss the combination of all the parts into a functional system with function that can be implemented. This includes the design of user interface, data flow and modules for the project.

Chapter 6, **System Implementation**, discuss the system development which includes the translation of the modules and algorithms to instruction according the programming language that is used.

Chapter 7, **System Testing**, discuss the testing process on the system to verified and validate the system according to the requirement and specification.

Chapter 8, **System Evaluation and Conclusion**, discuss the results, problems and solutions, advantages and disadvantages of the system developed, the enhancement for the system in the future, suggestion and also the conclusion for the project.

Chapter 9, **Appendix**, which is an additional chapters for illustration, original data and also questionnaire forms.

1.8 Project Schedule

The schedule for the development of CAS is shown in the figure 1.1.

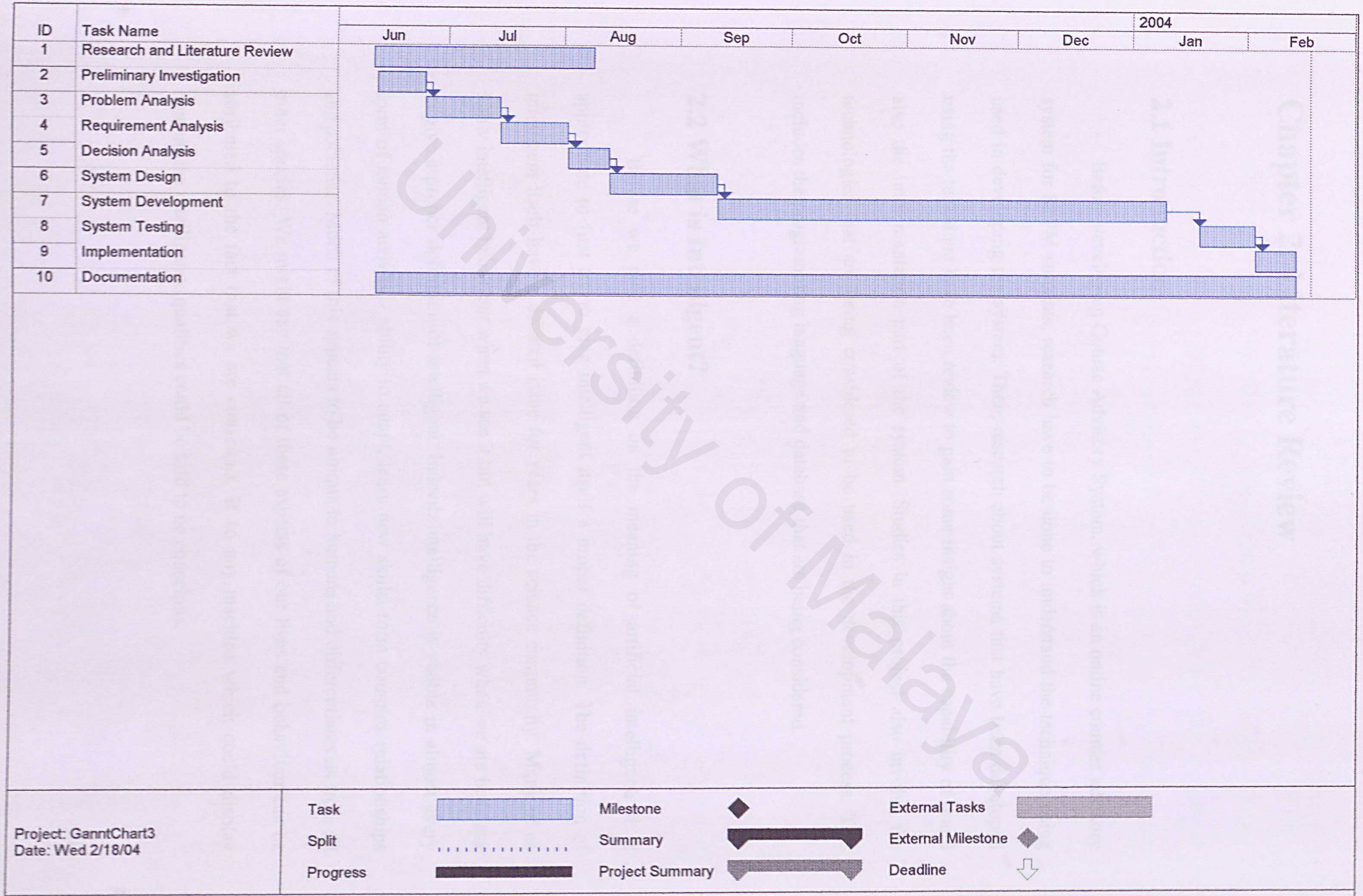


Figure 1.1: Project Schedule

Chapter 2: Literature Review

2.1 Introduction

Before developing Course Advisory System, which is an online courses advisory system for STPM students, research have to be done to understand the technique being used in developing the system. Then, research about systems that have been developed using the technique have been review to gain some insight about the capability of it and also the implementation part of the system. Studies in this project also involve the technologies that are being considered to be used in the development process. This includes the programming language and database that are being considered.

2.2 What is Intelligent?

Before we give a definition on the meaning of artificial intelligent, it is appropriate to first try to give intelligent itself a proper definition. The definition of intelligent itself have a debated issue for years in the science community. Many of us know intelligent behaviour when we see it but will have difficulty when we are been ask to give a proper definition of intelligent. Indeed, intelligence is visible in almost every form of human activity - ability to adapt, learn new skills, form complex relationships and societies. Much of this appears to be unique to humans and differentiates us from all other species. We might say that all of these aspects of our lives and behaviour can be attributed to the fact that we are *conscious*. If so any machine which could display human-like intelligence qualities could be said to be conscious.

This point of view was taken by **Alan Turing**, who in 1950 invented a test whose result could be used to determine whether, in any practical sense, a machine could be said to be conscious or intelligent. The test is quite simple. You enter a room and encounter two terminals: one terminal connects with a computer and the other interfaces with a person who types responses. The goal of the test is for you to determine which terminal is connected with the computer. You are allowed to ask questions, make assertions, question feelings and motivations for as long as you wish. If you fail to determine which terminal is communicating with the computer or guess that the computer is the human, the computer has passed the test and can be said to be `conscious'.

Turing invented his test at a time when it was thought that mind-like computers might be only fifty years away. A whole new science was born with the aim of producing such intelligent machines - the subject of artificial intelligence or AI. [2]

2.3 What is Artificial Intelligent?

Artificial Intelligence is a branch of *Science* which deals with helping machines finds solutions to complex problems in a more human-like fashion. This generally involves borrowing characteristics from human intelligence, and applying them as algorithms in a computer friendly way. A more or less flexible or efficient approach can be taken depending on the requirements established, which influences how artificial the intelligent behaviour appears. [3]

AI is generally associated with *Computer Science*, but it has many important links with other fields such as *Maths*, *Psychology*, *Cognition*, *Biology* and *Philosophy*, among many others. Our ability to combine knowledge from all these fields will ultimately benefit our progress in the quest of creating an intelligent artificial being.[3]

Artificial intelligent, unlike common believe especially in our country where A.I. is considered as an undemanded filed, is actually have a very high commercial value. A lot of artificial intelligent guideline are used virtually everyday, whether it be virtual assistants, firewalls, grammar and spell checking, etc. Remember that at the core, A.I. is about intelligent programs and machines. Not all applications may exhibit learning or adaptation; however, the execution of a pre-defined set of instructions allows them to act intelligently.

Additional A.I. based solutions can also include:

- Failure Prediction for network management solutions
- Fraud Detection
- Behavioural Analysis for market research
- Anti-Virus Protection
- Auto-System Analysis and Repair

Some of the research fields and applications that used in the artificial implementation are as follows [4]:

- Artificial Neural Network
- Evolutionary Computation / Artificial Life

- Cellular Automata
- Natural Language Processing
- Image Processing
- Game Playing
- Automated Reasoning and Theorem Proving
- Expert Systems
- Planning and Robotics
- Machine Learning
- Genetic Algorithms
- Case Based Reasoning

2.4 Introduction to CBR

Over the past few years, case based reasoning (CBR) has been an active field for research and there is also various implementation of CBR in solving real life problems. Thus, it is important that we have an in depth look at the meaning and methods in CBR.

Case based reasoning (CBR) means reasoning based on remembering previous experience on the problems and solutions. A reasoner will used old experience (cases) to suggest solutions for problems, to point out potential problems that might have occur using a solution being computed, making interpretation of a new situation and also making prediction on the outcome of that situation, or to create arguments justifying some conclusion. CBR uses an explicit database of problem solutions to address new problem solving situations. A CBR reasoner can solve new problems by interpreting the new situations by remembering old similar situations and comparing and contrasting the new one to old one to see where it fits best. Old situation can be obtained from human experts through the knowledge engineering process or may reflect the results of previous search-based successes or failures. For examples, medical education does not rely solely on theoretical models of anatomy, physiology and disease, but it also depends heavily on case histories and the intern's experience with other patients and their treatment.

CBR is a method that combines reasoning with learning where CBR learn by remembering much like the way expert reason. It spans the whole reasoning cycle. Each situation is considered as an experienced. It used the old situation stored to have a better understanding of the new situation. If there is a solution for the new situation from the old situation, the solutions will be applied in. Then the new situation along with the solution is inserted into the memory with the others cases to be used in future reasoning.

As we can see from the explanation above, the key aspect of CBR is remembering. Remembering in CBR can be divided into 2 parts. The first part consist of is storing the cases or experiences into the memory. The second part consists of retrieving the cases in the future to solve new problems. The CBR community calls this related set of issues **the indexing problem**. In broad terms, it means finding in memory the case closest with the current situation and applied the solutions of the previous to the current case. In a more specific term, CBR can be considered as a 2 part problem:

- Indexing or labelling the old or solved experiences when it is stored into the memory that describe the situation to which they are applicable so that it can be retrieve for later used.
- New situation are elaborated in detail in order for the indexes in the memory that are similar to it can be retrieve to solve the problem.

An acronym of CBR in human perspective is where lawyers select past law cases that are similar to their client's which suggest a favourable decision, and used the cases to convince the court that these similarities merit findings is in their client's favour. The other similarity in law principle with CBR is where the interpretations by the courts are usually based on legal precedents from previous conclusions. Cases later presented will be bound to the decisions made before. All these similarities of the law discipline and CBR has made CBR an ideal technique to be implemented in the justice system. (Kolodner, 1993)

2.5 What is a case and why we use case?

According to Janet Kolodner a case can be define as:

"A case is a contextual piece of knowledge representing an experience that teaches a lesson fundamental to achieving the goals of the reasoner."

This means that case in case based reasoning is situation that has occur before in the reasoner case base. It represent an experience situation that when remember will form a context in which knowledge embedded in the case is presumed applicable. Cases can be considered as specific knowledge that is tied to some specific situation, representing knowledge at an operational level. With this, they make explicit how task is carried out or how a piece of knowledge is applied or what particular strategies for accomplishing a goal were used. A case may come in different shapes and sizes, covering large or small time slices, associating solutions with problem, outcomes with situations, or both. It records experiences that are different from what is expected. Not all differences are important to record. Cases worthy of recording as cases teach s useful lesson. Useful lesson are those that have the potential to help a reasoner achieve a goal or asset of goals more easily in the future or that warn about the possibility of a failure or pint out an unforeseen problem. (Kolodner, 1993)

There are a numbers of advantages of using cases in problem solving rather than using general knowledge as given below as well as problem that occur using general knowledge of reasoning:

- Operationalization problem in using general knowledge where it is sometimes difficult to applied it to a particular situation.

- Rules in general knowledge sometimes is express too abstract.
- General knowledge only try to cover the normal where it cause the lack of understand on situation that is out of norm especially if there is lack in the knowledge.
- Cases are able to capture knowledge that might be too hard to be capture in general mode.
- Cases chunk together knowledge that belongs together.

2.6 CBR Methods

Central tasks that all CBR methods have to deal with are to identify the current problem, find similar past case, used to suggest solution, evaluate solution and update system by learning. How this is done, what part of the process that is focused, what type of problem that drives the method, etc. various considerably, however. Below is an attempt to classify CBR methods into types with roughly similar properties in this aspect.

2.6.1 Main Types of CBR Methods

The CBR paradigm covers a range of different methods for organizing, retrieving, utilizing and indexing the knowledge retained in the past cases. Cases maybe kept as concrete experiences or a set of similar cases may form generalized case. Cases may be stored as separate knowledge units or split up into subunits and distributed within the knowledge structure.

Case maybe indexed by a prefixed or open vocabulary, and within a flat on hierarchical index structure. The solution from a previous case may be directly applied to the present problem or modified according to different between the two cases. The matching of cases, adaptation of solution and learning from experience may be guided and supported by a deep model of general domain knowledge; by more shallow and compiled knowledge or be based on an apparent, syntactic similarity only. Case based reasoning methods may be purely self-contained and automatic or they may interact heavily with the users for support and guidance of its choice. Some CBR method assumes a rather large amount of widely distributed cases in its case base, while other are based on a more limited set of typical ones. Past cases may be retrieved and evaluated sequentially or in parallel.

Actually, Case base reasoning is just one of a set of terms used to refer to systems of this kind. This has lead to some confusion, particularly since CBR is a term used both as a generic term for several types of more specific approaches as well as for one such approach. To some extent, this can also be said for analogy reasoning. An attempt of a classification, although not resolving the confusion, of the term related to CBR are as follow:

- **Examplar-based reasoning**

  The term is derived from a classification of different view do concept definition into "the classical view", "the probabilistic view" and "the examplar view". In the examplar view, a concept defined extensionally, as the set of its examples. CBR methods that address the learning of concept (i.e. the problem addressed by most of the research in machine learning) are sometimes referred to as examplar-based. In this approach, solving a problem is a classification task, i.e. finding the right class for the unclassified examplar. The class of the most similar past case becomes the solution to the classification problem. The set of classes constitutes the set of possible solution. Modification of a solution formed is therefore outside the scope of this method.

- **Instance-based reasoning**

  This is a specialisations of examplar-based reasoning into s highly syntactic CBR-approach. To compensate for lack of guidance from general background knowledge, a relatively large number of instances are modelled in order to close in a concept of definition. The representations of the instance are usually simple (e.g. features vector), since a major focus is to study automated learning with no user in the loop. Basically, this is a non-generalization approach to the concept learning problem addressed by classical, inductive machine learning methods.

- **Memory-based Reasoning**

  This approach emphasized a collection of cases as a large memory, and reasoning as a process accessing and searching in this memory. Memory organization and access is a focus of the case based methods. The utilization of parallel processing technique is a characteristic of these methods and distinguishes this approach from the others. The access and storage methods may rely on purely syntactic criteria, as in the MBR-Talk system or they may attempt to utilise general domain knowledge.

- **Case-based Reasoning**

  Although case based reasoning is used as a generic term in this paper, the typical case based reasoning methods have some characteristics that distinguish it from the other approaches listed here. First, a typical case is usually assumed to a certain degree of richness of information contained in it, and a certain complexity with respect to its internal organization. That is a features vector holding some value and a corresponding class is not what we would usually call a typical case description. What we refer to as a typical case based methods also have another characteristics property. They are able to modify or adapt a retrieved solution when applied in a different problem solving context. Paradigmatic case based method also utilizes general background knowledge – although its richness, degree of explicit representation and role within the case based reasoning processes varies. Core methods of typical case based reasoning system borrow a lot from cognitive psychology theories.

- **Analogy-based Reasoning**

  This term is sometimes used as a synonym to case based reasoning. However, it is also often used to characterize methods that solve new problems based on post cases from a different domain while typical case based reasoning methods focus on indexing and matching strategies for single-domain cases. Research on analogy reasoning is therefore a sub-field concerned with mechanisms for identification and utilization of cross-domain analogy. The major focus of study has been on the reuse of a post case, what is called the mapping problem: finding a way to transfer or map the solution of an identified analogue (called source or base) to the present problem (called target).

The fact that s system is described as an example of some other approach does not exclude it from being a typical case based reasoning system as well. To the degree that more special examples of, e.g. instance-based, memory-based or analogy-based methods will be discussed, this will be stored explicitly. (Goh Shu Kea, 2002)

2.7 Case based Reasoning Cycle

CBR is a technique that based on getting the closest match between the current problems with those that are stored to find the solution. Although they are a lot of way to implement CBR, they all still shared a common structure. The reasoning cycle consist of retrieving, modifying, applying and finally saving the solution. Figure shows the CBR cycle:

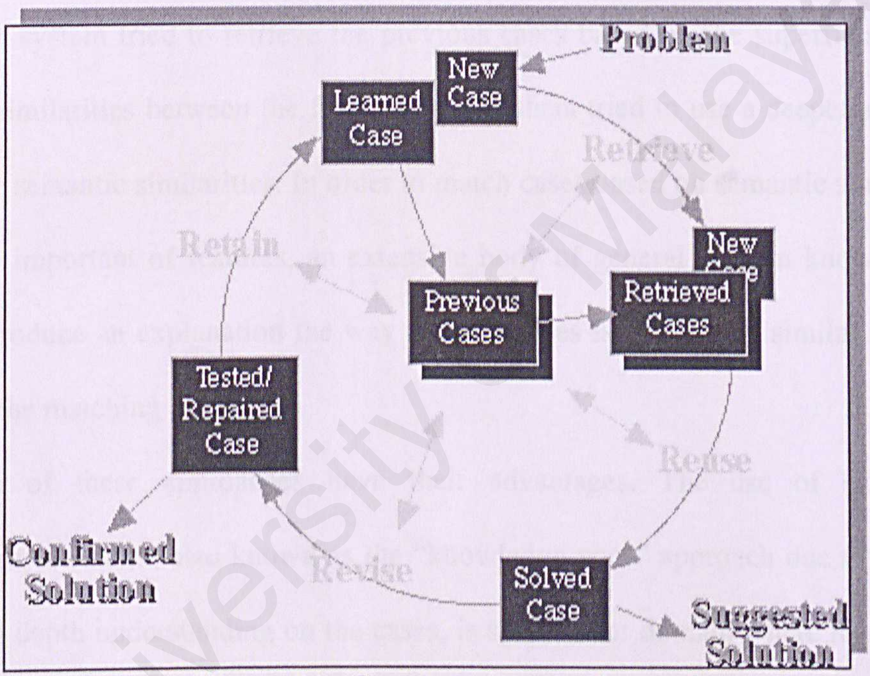


Figure 2.1: Case based reasoning cycle

A detail of each steps involved in the reasoning process are as follow:

2.7.1 Case Retrieval

Case retrieval is one of the primary processes in the case based reasoning cycle where cases that have features similarity with the current problem will be retrieved. Case retrieval can be divided into a few subtasks namely features identification, initial match, search and select. Features identification is primary used to identifying and coming up with problem descriptors that it the important features in the current problem. Next, the features will be match with the cases in the case base and cases with high similarity will be return. Finally, from the cases that have been match, the best match case will be chosen as solution for the current case (or at least first case to try out).

There have been a few approach used in retrieving the case in the case base. Some of the system tried to retrieve the previous cases based on the superficial or the syntactical similarities between the features while others tried to use a deeper approach by using the semantic similarities. In order to match cases based on semantic similarities and relative important of features, an extensive body of general domain knowledge is needed to produce an explanation the way the two cases is considered similar and also how strong the matching is.

Both of these approaches have their advantages. The use of syntactical similarities of the cases, also known as the "knowledge-poor" approach due to the lack having an in depth understanding on the cases, is suitable for domain where it is hard or impossible to extract the general knowledge out. On the other hand, semantic similarities approach, also known as the "knowledge-intensive" approach as it has certain knowledge on the cases, are able to use the contextual meaning of a problem descriptors in its matching, for domain where general knowledge is available.

Given below are the explanations of the subtasks that are involved in the case retrieval process.

Features Identification

As been explained above, some systems uses the syntactical similarities (knowledge-poor) approach will simply identified the input features from the current case while those that use the semantic similarities approach (knowledge-intensive) will attempt to 'understand' the problem and features within its context. Features that are unknown to the system may be disregarded or it may ask back the user for an explanation on it. In PROTOS, the second approach of the knowledge-intensive approach has been used where the system will ask the user to provide an explanation on features that are unknown to the system where the explanation provided should be able to link the features in to the existing semantic network (category structure). The explanation is used to understand problems features, to check whether the features value make sense within the context, to generate expectation of others features. Other descriptors than those given as input, may be inferred by using a general knowledge model, or by retrieving a similar problem description from the case base and use feature of that case as expected features. Checking of expectation may be done within the knowledge model (case and general knowledge), or by asking the user.

Initially Match

After identifying the features in the current case, the next step is to find cases in the case base that match the identified features. This task can be further divided in to two steps which are the initial matching process that will retrieve a set of cases that have similar features with those that have been identified and a more elaborate process of finding the best case among those cases in the retrieved set. The case that is selected should be suitable to solve the problem. The latter process is called the select task. Set of matching cases from the case base can be retrieve by either using the problem features as in the indexes to the case base either directly or indirectly. There are 3 principle ways of retrieving a case from the case base. The first is by following direct index pointers from problem features. Second is by searching an index structure and finally by searching in a model of general domain knowledge. The first strategy has been implemented in PATDEX for its diagnostic reasoning, and also the second strategy in the test selection. A domain-dependent, but global similarity metric is used to assess similarity based on surface match.

Cases may be retrieved solely based on the input features, or it can also from features that are inferred from the input features. A complete matching of the features is, of course, a good candidate to be used but it also depend on the strategy that is used. Cases that match a given features of the problem features may also be retrieved. Some tests may be executed to test for the relevance of a case, particularly if the cases are retrieved on the basis of a subset of features. A way to access the degree of similarity is needed, and several 'similarity metrics' have been proposed, based on surface similarities of problem and case features.

Similarity assessment may also be knowledge-intensive oriented by understanding the problem more deeply by using the goals, constraint, etc. from this elaboration process to guide the matching. Others option is by weighting the problem features according to their importance in the problem.

Select

Selecting is the process that is used to find the best matching case from those cases that have been retrieved from the initial matching process. Although this process could be done during the initial matching process, but it is best left separated as the initial matching process tend to return a set of similar cases. The best case from the set is normal selected by further evaluating the degree of initial match more closely. This is done by an attempt to generate explanation to justify non-identical features, based on the knowledge in the semantic network. When a case turn out to be not strong enough, an attempt to find a more appropriate case is made by following difference links to closely related cases. The selection process normally will try to generate some consequences and explanation from each retrieval cases, and attempt to evaluate them. This could be done by using the system's own general domain knowledge or it could ask the user for some external help for confirmation and addition information. The cases later will be ranked according to some metric or ranking criteria. Knowledge-intensive system selection method will tend to generate explanations that support its ranking process. Case that has the strongest explanation will be chosen to solve the new problem. Other properties that can be considered during this process are the relative importance and discriminatory strengths of features, prototypically of a case within its assigned class and difference links to related case.

Retrieval Method

Nearest Neighbour method

This approach uses a weighting sum of features between the stored cases and the new input case for assessment of similarity between them. The equation of a typical algorithm for calculating nearest neighbour matching by Cognitive System Remind software is given below:

$$\sum_{i=1}^{n} \frac{w_i \times sim(f_i^{I}, f_i^{R})}{\sum_{i=1}^{n} w_i}$$

where w_i is the importance of dimension (slot) I, sim is the similarity function for primitives, and f_i^{I} and f_i^{R} are the values for features f_i in the input and retrieved cases respectively.

The biggest problem using this method is to determine the weight of the features. The limitation of this approach includes problem in converging on the correct solution and retrieval time. In general the use of this method will leads to the increasing in the retrieval time as the case base grow larger. This method is suitable for system that has a limited case base as in the BROADWAY system.

Induction

The induction algorithm is used to determine which features is best used to discriminating cases and generate a decision tree type structure to organize the cases in memory. This approach is useful when a single case feature is required to be used as a solution and where that case feature is dependent upon others.

Knowledge Guided Induction

Knowledge is applied to the induction process by manually identifying case features that are known or though to effect the primary case feature. This approach is frequently used in conjunction with other technique, because the explanatory knowledge is not always readily available for large case base.

Template retrieval

This approach is similar to SQL-like queries in the database environment. Template retrieval returns all cases that fit within certain parameters. This technique is often used before other technique to reduce the searching time and also the scope of searching.

2.7.2 Case Reuse

The focus of case reuse after it has been retrieve is on two aspect which are the difference between the past case and the current case and which part in the past case should be transfer in to the current case as solution.

Copy

This method will obstruct away differences that have occurred between the two cases. This means that any differences in the two cases are considered as irrelevant and any similarities are considered relevant and the solution class of the retrieved case will be transferred to the current case as the solutions class. This is trivial type of reuse. However, other system have to take in to account differences in the cases thus cannot be directly transferred to the current case but it requires an adaptation process that will takes those differences in to account.

Adapt

There are two main ways to reuse past cases. The first one is to use the past case solutions as the solution for the new case. This way is called the transformational reuse. The other ways is called derivational reuse. In derivational reuse, past method that is used to solve the case will be reapplied in to the new case instead of the solution of the past case. In transformational reuse, the solutions in the past case is not the exact solution for the new case, but it did contains some knowledge in the form of transformational operators {T} such that applied to the old case, they will transform it in to a solution for the new case. A way to organize these transformational operators is to index them around the differences detected among the retrieved and current case. Transformational reuse does not look at how a problem is solved but focused on the equivalence of solution and this require a strong domain-dependent model in the form of transformational operator {T} plus a control regime to organize the operators application.

However, derivational reuse looks a how the problem is solved in the past case. The past case contains information of how a problem is solved in the retrieved case. These include a justification of the operators used, subgoal considered, alternatives generated, failed search path, etc. Derivational reuse then reinstantiates the retrieved methods in to the new case and 'replay' it. During the replay, successful path will explore first and those paths that lead to failure will be avoided.

2.7.3 Case Revision

When solution that is generated by the system is incorrect, there is a need to revision the past case that is required. This revision process can be considered as part of the learning process of the system. Case revision consists of two parts that is evaluation of the solution that has been used and case repair. When the evaluation process is successful, the system will learn from the success and there is no need to repair the case but if its fail, then the case solutions will be repaired using some domain-specific knowledge.

The solution evaluation task will take the result from applying the solution in the real environment (by asking a teacher or performing the task in the real world) to evaluate the effectiveness of the solutions. There is also system that use simulator of the real world to evaluate the solution as in the CHEF system.

The case repair task involved the detection of the error in the solutions that has been applied and generating an explanation for them. An example is the CHEF system where it uses a casual knowledge is used to generate an explanation of the reason for the failure to achieve its goal. CHEF will learn the general situation that leads to the failure by using an explanation-based learning technique. This is included in to a failure memory that is used in the reuse phase to predict possible shortcomings of its plan. This is a form of learning where it moves the detection of errors in a past hoc fashion to the elaboration phase where error can be predicted and avoided. The second task is the solutions repair tasks where solutions are modify using the failure explanation so that future error can be avoided. One of the ways is to incorporate an repair module that possess general causal and domain knowledge about how to disable or compensate causes of errors in the domain.

2.7.4 Case Retainment- Learning

This is idea of this process is to retain the new problem solving episode in to the case base as part of the learning process. The system not only retains those cases that are successful but also those that have failed. The steps include selecting information from the case to retain, the form of the information be retain in, the indexing of the case and also the integration of the case in to the memory structure.

Extract

In case based reasoning, case are updated in to the case based regardless the way the case is solve. If the current case is solve by using a previous case, it can either built a new case for the current case to be updated in to the case base or it can generalize the previous case to subsume the present case as well. If the case is solve by other methods such as asking the user or pass it to another reasoning system, an entirely new case will have to be constructed. An important question is what to use as the source of learning.

Relevant problem features and problem solutions are obvious the candidate but an explanation or other form of justification on why a case is considered as a solution to the problem may also be marked for inclusion in a new case.

Index

Indexing has always been a major issue in case based reasoning. It amount to deciding on the types of indexes to be used for future retrieval and the structure of the search space of the indexes. As been mentioned previously, direct indexing, skips the latter steps but there is still the problem of identifying what types of indexes to use. One of the ways is to use all the input features as the indexes. This is the approach of syntax-based methods within instance-based and memory-based reasoning. In the memory-based method of CBR-Talk, for example, relevant features are determine by matching, in parallel all cases in the case base and filtering out features that belong to cases with few features in common with the current case.

Integration

Integration is the final steps that are involved in updating the case base with the new case. If no new case or index has been constructed, then integration is the main step of Case retainment. By modifying the indexing of the existing case, case based reasoning system learns to become better at assessment the similarity. The tuning of existing indexes is an important part of the learning process. Indexes strength or importances for a particular case or solution are adjusted due to the success or failure of using the case to solve the input problem. For features that have been judged relevant for retrieving a successful case, the association with the case is strengthened, while it is weakened for features that lead to failure. In this way, the index structure has a role of tuning and adapting the case memory to its use. (Kolodner, 1993; Goh Shu Kea, 2002)

2.8 CBR software tools

With the increasingly interest in case based reasoning in recent years, it is no wonder that a lot of tools have emerge and commercially available to help in developing case based reasoning system. Some of these tools are presented here together with a summarization of its functionality.

ART*Enterprise

ART*Enterprise is product of Inference Corporation. Inference Corporation based in California is one of the oldest established vendors of AI tools. Inference market ART*Enterprise as an integrated, object-oriented applications development tool designed for MIS developer's offering a variety of representational paradigms including:

- a procedural programming language
- objects supporting multiple inheritance, encapsulation and polymorphism;
- rules;
- cases.

Included with this package are a GUI builder, version control facilities, and an impressive ability to link to data repositories in most proprietary DBMS formats for developing client-server applications. Moreover, ART*Enterprise offers cross-platform support for most operating systems, windowing systems and hardware platforms.

The CBR component in ART**Enterprise* is essentially the same as that in CBR Express (or rather vice-versa since CBR Express uses code from ART to provide its CBR functionality). This functionality is reviewed below in Section 0. However, because developers have direct access to the CBR functionality ART**Enterprise* is more controllable than in CBR Express.

ART*Enterprise is the ideal tools in implementing CBR functionality for corporate wide information system. Although the CBR functionality itself is more limited than some tools, the proven knowledge representational abilities of ART will make it a good tool for performing complex case adaptation. It can be assumed that since ART*Enterprise uses similar code to CBR Express that its case retrieval times will be as fast (or faster) than those recorded in Althoff et al's experiments [95]. It is recommended that ART*Enterprise run on a fast 486 or Pentium with a minimum of 32 MB of RAM as it is a very demanding on the resources especially when it is run on PC platform under MS Windows.

CasePower

CasePower is developed by Inductive Solutions Inc. Previously it is known as Induce-it. Cases are built within the spreadsheet environment of Microsoft Excel making it a specialised tool for constructing Excel spreadsheets that can be analysed using CBR. With the use of Microsoft Excel, it contains all the features that are available in Excel such as graphing, reporting and DDE. Within the limited confines of Excel, it provides basic CBR functionality mainly suitable for numeric applications. Symbolic data can be represented as ordered hierarchies that are mapped to numerical ranges. However, for more complex non-numerical applications another CBR tool may be preferable.

The retrieval method that is used in this tool is the nearest neighbour method. One of the interesting features is its ability to reduce search time by calculating an index for each case in advance but this features is not too suitable to be used on system that has a large case base as it can become a lengthy process. The system simply calculates the index for the new case and compares it against the pre-calculated indices of the case-base. If a new case is to be retained, the entire set of case indices must be recalculated. Adaptation can be performed using Excel formulae and macros.

ReCall

ReCall is a CBR trademark of the Paris based AI company ISoft. This tool offers a combination of nearest neighbour and inductive case retrieval. ReCall is coded in C++ and is available on the PC under Windows 3.1, on UNIX Workstations. It is designed on an open architecture allowing users to add CBR functionality to their own applications.

Recall presents an object-oriented language with taxonomies, inheritance and multiple-inheritance mechanisms, typed descriptors, facets, deamons, and relationships between objects (individual cases are represented as instances). This allows users to specify complex domain knowledge in a structured but modular way, and to describe cases having noisy, incomplete and uncertain descriptions since feature values can be inherited. Recall provides multiple hierarchical indices that are used for organisation purposes and for efficient case retrieval. ReCall provides different methods for automatically analysing the case base providing for selection of indices as well as their organisation. However, experienced developers can impose their own organisation. Automatic procedures are based on inductive techniques. The automatic procedures takes into account the domain knowledge defined in the cases, helping users to develop applications interactively. Similarities functions take into account both the properties and values of descriptors, as well as structural differences between cases. ReCall uses a variant of a nearest-neighbour algorithm that improves similarity computations.

ReCall supports two different adaptation mechanisms: a default adaptation mechanism based on a voting principal, and user defined adaptation rules. As ReCall is based on C++, external function calls can provide more complex adaptation. ReCall can be interfaced to external applications in particular with data bases and since ReCall is available as a C++ library, CBR functionality can be integrated with other applications.

Through the use of specialised graphic editors, the developer can define objects, relationships between objects, taxonomies, deamons and adaptation rules. A tree editor allows the user to interact directly on the case organisations in order to control and modify indices. A user mode allows developers to write adaptation rules or daemons, whilst a developer mode gives access to an interpreted language.

S3-Case

S3-Case is developed by techInno's S3 as an environment for systems maintenance running on PC Windows, Mac, OS/2 and various UNIX platforms. It is written in SMALLTALK where it support object oriented model with inductive and nearest neighbour retrieval and adaptation methods. Rule can also be used to prune the search space before retrieval. A simple user interface can be customised to suit user needs. Those who are experienced in developing using SMALLTALK can embed or extend the functionality of S3-CASE.

2.9 Advantage and Disadvantage of using Case Based Reasoning

As case based reasoning is derived from the way human reason base on experience, there are a numbers of advantages that appeal it to be applied as a problem solving method. Some of the advantages of using case based reasoning are given below:

- Case based reasoning allows reasoning to be done quickly to purpose a solution as it used previous solutions rather than reason it from the scratch.

  Example: Doctor remembers previous diagnostic and used it to diagnose current patient and uses the solution of the previous case. This can save a lot of time if the symptom of the patient has similarity with some previous patient.

- Case based reasoning allows the reasoning to be done on a domain that it does not completely understand.

  Example: Using case based reasoning in a domain that the system does not completely understand is based on the idea that what works previous should have works now.

- Case based reasoning gives a reasoner a means of evaluating solutions when no algorithmic method is available for evaluation.

  Example: Using cases to aid in evaluation is particularly helpful when there are many unknowns, making any other kind of evaluation impossible or hard. Solutions are evaluated in the context of previous similar situations. Again, the reasoner does its evaluation based on what worked in the past.

- Cases are useful when interpreting concepts that are open-ended and ill-defined.

  Example: This concept is very important used by attorneys as well as daily situation.

- Case based reasoning is very useful in detecting potential problem and gives warning for problem that has been occurred in the past.

  Example: By using error that has occur during the past case, case based reasoning can use it to prevent such error to occur in the future so it is very important for case based reasoning system for not only remembering successful cases but as well as those cases that have failed.

- Case based reasoning can help the reasoner to focus on the important parts of the problem.

  Example: This is based on the concept that features in the previous case that are important tend to be also important in the current case. This is true as features that can lead to the success solving of the problem can be used to solve the current problem. Features that lead to failure in the previous cases should be used to be avoided.

As with all human way of reasoning, case based reasoning is also prone to having a numbers of disadvantages as mentioned below:

- A case based reasoner may be tempted to use old cases blindly, relying on previous experience without validating it in the new situation.
- Bias can happen when solving a new problem

2.10 Case based reasoning in comparison with other reasoning method

Case based reasoning has proven to be an effective way to be used in computational reasoning method. With advantages like reducing the reasoning time needed, ability to warn of potential problems and give reasoner to notice opportunity base on what worked at the past, case based reasoning has become to choice in many reasoning system. In fact, there has been evidence that case based system has outperformed some traditional expert system. For example, CASEY has been proven to be as accurate as the Heart Failure Program, a model-based program that uses the same knowledge to reason, and two to three orders of magnitude faster than it. PROTOS, which diagnoses hearing disorder, was 50 present more accurate than other classifier in an evaluation study (Porter, Bareiss, and Holte 1990). A comparison of case based reasoning with rule-based and model-based reasoning is given below:

2.10.1 Case-based and Rule-Based reasoning

A major difference in these two reasoning methods is the emphasis in the research area. In rules based reasoning, the emphasis is more on the mechanism of reasoning and the form of the knowledge, with less emphasis on the content of the knowledge. The knowledge is encoded in to the rules while there is little guidance given about the content of those rules should be. While in case based reasoning, majority emphasis is given to the issues of content. These issues include the kind of content the case should have, the adaptation strategies being use and the knowledge that needed for the system to work. The table 2.1 shows some others differences between the two reasoning methods:

Table 2.1: Comparison between case base reasoning and rule based reasoning

Case based reasoning	Rule based reasoning
Cases are constant	Rules are pattern
Cases retrieve match partially	Rules retrieve match exactly
Cases are first retrieve, approximating the entire solutions at once, then adapted in the solutions	Rules are applied iteratively in micro events
Cases are large chunks of domain	Rules are small, ideally independent but

knowledge, quite likely redundant, in part, with other cases	consistent pieces of domain knowledge.
Cases are real situation that is easier to be acquire.	Knowledge are acquire from the expert and encoded in to rules
Case based show explanation by showing cases its solution derived from	Rule based gives rule chains as explanations.

2.10.2 Case based and Model based reasoning

Both case based reasoning and model reasoning had shown a numbers of similarity between them. Both of them were developed as a method to bypass the need to reasoning from the scratch and also both compose knowledge into large chunks and reason using large chunks.

The major differences between both reasoning are given in the table 2.2:

Table 2.2: Comparison between case base reasoning and model based reasoning

Case based reasoning	Model based reasoning
Cases store describe the way things work	It sore causal models of devices or domain
It provides for efficient solution generation, and evaluation is based on the best cases available	It provides mean of verifying solutions but solution generation is not guided.
It can be applicable to the same condition as model based plus also in domain that is not understood well.	It is applicable if there is a causal model available that is when the domain knowledge is well understood.

Although these two reasoning method have its differences, it is interesting to notice that both it is also complement with each others. While model based reasoning tend to hold knowledge needed for validation or evaluation of solutions but it do not provide methods for constructing solutions. In case based reasoning, a means of evaluation is needed to evaluate the solutions, guiding its adaptation, and knowing when two cases are similar and the general knowledge in model based reasoning are suitable to provide that. In fact there have always been system developed that integrate both methods in it. Some of the examples are CASEY and KRITIK. (Kolodner, 1993)

2.11 Example Systems that use the Case Based Reasoning method

Over the years, a lot of research have been done on exploring the potential of case based reasoning has spark the interest in developing system that has implement case based reasoning as the problem solving method. Case based reasoning has been successfully implemented in a lot of fields such as law, medicine, designing, planning, classification and also diagnostic. Some examples of these systems are given below together with a brief introduction on the system:

Another Academic Advisor

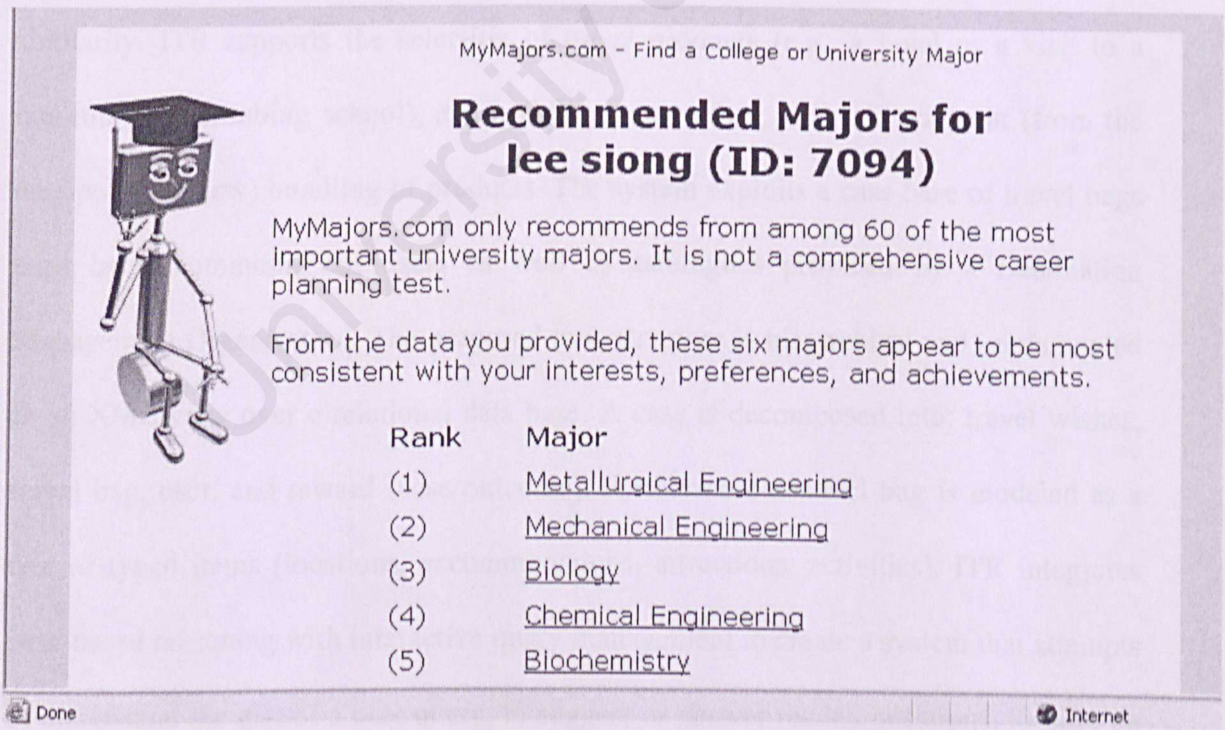
Another academic advisor (AAA) is a system developed by Binh Viet Nguyen from Ohio University. The objective of the system is to help human academic advisor in selecting suitable courses for students on probation. AAA works on the basis of that similar student will have similar academic performance. It makes course recommendation based on the academic performance of the student in the past and courses that other similar students has succeeded in.

In Ohio University, students are considered in probation if they earn one or more deficiency point (the accumulative grade point average falls below 2.0). The students will be removed from the probation list if they eliminated all deficiency points (earned a 2.0 accumulative grade point average - GPA). Students that are under probation will have to get advice from their academic advisor on their courses selection. Previously, academic advisor have to rely on their experience in giving advice to students. AAA is use during this period in order to help academic advisor to plan for courses that are suitable for the students to get a high grade.

AAA used the Nearest Neighbour Algorithms when matching for the most similar case in the case base. Cases in the case base are ranked according to their similarity and case that has the highest ranking is selected as solution. The lists of courses are divided in to two lists. The first list contains those courses that are suitable for the students and have a high percentage in helping students to get a good result. The second list is in contrast with the first list where it contains those courses that the students should avoid to take. [9]

www.mymajors.com

Mymajors.com is a web-based application constructed by Current Software that is use to advice students regarding the majoring that they should take in the university or college. The system is intended to be use by high school senior and also college freshman and use by the United States of America students. The users will be asked to complete a 15 minute interview to gain some data of the students. Some of the questions being asked are the students' result including their specific result on the courses, the students' preferences where the student input those majoring that they are interested in, majoring that the students wish to avoid and their interest in the subject they took in the high school. These questions are emerged from interview with actual academic advisors. Based on these input, the system will try to look for majoring that suit with the user input. After analysing those inputs, the system will recommend 6 majoring to the students. These majoring are chosen based on the students' interests, results and also their high school experience. Besides, the system can also be used to provide additional information on the majoring such as what the major is about, type of courses should take to complete the degree and the job opportunity. An example of the interface of the results is given in the figure below [5]:



MyMajors.com - Find a College or University Major

Recommended Majors for lee siong (ID: 7094)

MyMajors.com only recommends from among 60 of the most important university majors. It is not a comprehensive career planning test.

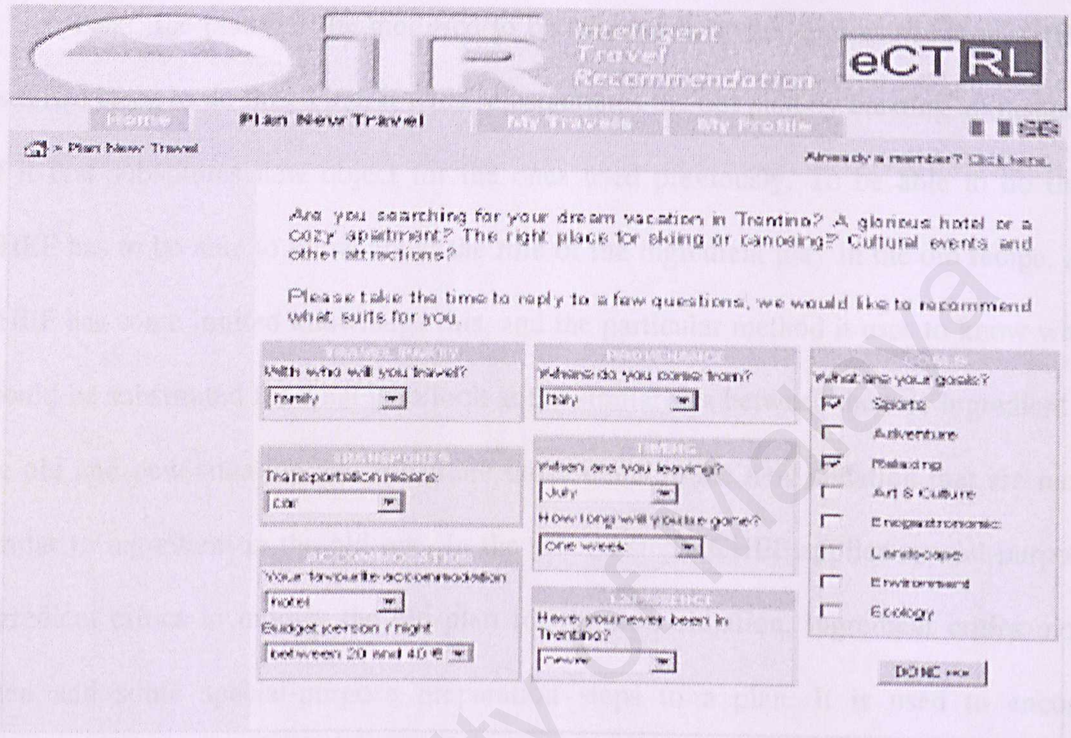
From the data you provided, these six majors appear to be most consistent with your interests, preferences, and achievements.

Rank	Major
(1)	Metallurgical Engineering
(2)	Mechanical Engineering
(3)	Biology
(4)	Chemical Engineering
(5)	Biochemistry

Figure 2.2: Interface showing the result from Mymajors.com

INTELLIGENT TRAVEL RECOMMENDER (ITR)

ITR is a web based recommender system aimed at supporting a user in information filtering and product bundling. It is developed by eCommerce and Tourism Research Laboratory using the case based reasoning approach. Recommender systems for travel planning try to mimic the interactivity observed in traditional counseling sessions with travel agents. The system enables the selection of travel locations, activities and attractions, and supports the bundling of a personalized travel plan. A travel plan is composed in a mixed initiative way: the user poses queries and the recommender exploits an innovative technology that helps the user, when needed, to reformulate the query. Travel plans are stored in a memory of cases, which is exploited for ranking travel items extracted from catalogues. A new 'collaborative' approach is introduced, where user past behavior similarity is replaced with session (travel plan) similarity. ITR supports the selection of travel products (e.g., a hotel or a visit to a museum or a climbing school), and building a travel bag, that is a coherent (from the user point of view) bundling of products. The system exploits a case base of travel bags built by a community of users as well as catalogues provided by a Destination Management Organization. The proposed case structure is hierarchical and implemented as an XML view over a relational data base. A case is decomposed into: travel wishes, travel bag, user, and reward (case outcome). Furthermore a travel bag is modeled as a tree of typed items (locations, accommodations, attractions, activities). ITR integrates case-based reasoning with interactive query management to create a system that attempts to understand the gist of a user query, to suggest or answer related questions, to infer an answer from data that is accessible, or to give an approximate response. ITR tries first to cope with user needs satisfying the logical conditions expressed in the user's query and, if this is not possible, it suggests query changes that will produce acceptable results. The figure shows the main interface for ITR [11]:



Are you searching for your dream vacation in Trentino? A glorious hotel or a cozy apartment? The right place for skiing or canoeing? Cultural events and other attractions?

Please take the time to reply to a few questions, we would like to recommend what suits for you.

With who will you travel? family

Where do you come from? Italy

What are your goals?
☑ Sports
☐ Adventure
☑ Relaxing
☐ Art & Culture
☐ Enogastronomic
☐ Landscape
☐ Environment
☐ Ecology

Transportation means: car

When are you leaving? July

How long will you be gone? one week

Your favourite accommodation: hotel

Budget person / night: between 20 and 40 €

Have you never been in Trentino? never

Figure 2.3: Main interface for ITR

CHEF

CHEF is a case based reasoning system that is used as a planner. The main objective of the system is recipe creation where it is view by CHEF as a planning process. They provide the sequence of steps that must be carried out to achieve the creation of some dish. The inputs in to the system are those goals that the system has to be able to achieve and the output of it is the recipe that is able to achieve those goals from the input. For example, the inputs are use stir-frying method and sweet taste.

To achieve this, CHEF will first recall old plans that worked under similar circumstances. It will look for plans that are able to fulfill the input as many as possible by indexing its plan by the goals they achieve.

Next, the plan will be modified to be adapted in to the current situation. CHEF uses two steps to do this. First, it wills reinstantiates the old plan by creating an instance of it that substitutes new object for the ones used previously. To be able to do this, CHEF has to be able to know about the role of the ingredient play in the old recipe. As CHEF has some limited knowledge this, and the particular method it uses to know what should be substituted for what is to look at the similarities between the two ingredient is the old and new situation and substitute those items in the new situation that are most similar to ingredient in the old one. In the second steps, CHEF applies special-purpose ingredient critics to modify the old plan for the new situation. Ingredient critics most often add some special-purpose preparation steps to a plan. It is used to encode knowledge on some procedure that is needed to prepare certain ingredient.

CHEF will then use its plan and collect feedback on the successfulness of its plan. If it is successful, then the procedure is considered finish. But if it is fail, CHEF will attempt to learn from its failure. Beside that, CHEF can update its understanding of the world so that it will be able to anticipate and not repeat the mistake it has just made. (Kolodner, 1993)

CASEY

CASEY is a case based diagnostician. The main objective of CASEY is to produce a causal explanation of the disorder of a patient from input such as normal signs and presenting signs and symptoms. Its use a model-based matching and adaptation heuristics to the cases it has available. CASEY has a case library of 25 cases, all of which were diagnosed by the Heart Failure Program (Long Et al, 1987) and it is also built on top of the Heart Failure Program. The program is a model-based diagnostic program that diagnoses heart failure with unprecedented accuracy.

When a new patient is presented to the system, it will first use CASEY to search through it case library to look for any case that have similarity with the patient. If a match is found, it will be used to diagnose the patient. But in the event of failure, that is if there is no match found, the case will be passed to the Heart Failure Program to be diagnosed and the result will then be sent back to CASEY to be stored for future used. There are two steps that are involved in case based diagnosis. First, CASEY will search it case library to find any similar cases that can be used by using the model-based evidence rules to determine which of the partially matching cases that are retrieved are sufficiently similar to suggest an accurate diagnosis. The second steps consist of applying a model-based repair rule (adaptation strategies) to adapt the old diagnosis to fit the new situation.

The evidence rule in CASEY is used to reconcile differences that occur in the retrieve case with the current case. It specifies under what circumstances differences can be reconciled. CASEY's evidence heuristic examine the role of each descriptor play in the previous diagnosis and the role all the new descriptors could play in the same diagnosis, and it attempt to match there outlines features to each other. If it is successful, the match is validated but if not the old case is discarded and other cases will be tried out.

As CASEY is built on top of the Heart Failure Program, it is able to get the accuracy from the program and also shown an impressive improvement in performance comparing to the program especially in term of speed. CAESY is a good example of showing that case based reasoning has a potential to speed up the model-based programs considerably without the loss of accuracy. (Kolodner, 1993)

2.12 Programming language being considered

2.12.1 ASP.NET

ASP.NET is the hosting environment that enables developers to use the .NET Framework to target Web-based applications. However, ASP.NET is more than just a runtime host; it is a complete architecture for developing Web sites and Internet-distributed objects using managed code.

A key feature of ASP.NET is the separation of code into a separate file from the HTML page that calls it. This 'code behind' concept helps clarify the roles of designers and developers, and neatly accommodates another important .NET technology, namely XML Web Services. From the coder's point of view, this makes it a lot easier to discern the logic behind a Web Form or Web Service (Payne, 2003).

Pros

- Data access, windowing, connecting to the Internet, and much of the functionality of the Win32 API is now accessible through a very simple object model.

- The VB language has been hugely upgraded, so it now includes classes and most of the features previously accessible in C++.
- A new language, C#, has been introduced, which combines the efficiency of C++ with some of the ease of development of VB.
- Memory management for .NET applications is much more sophisticated, meaning that a badly behaved .NET component is extremely unlikely to crash other components running in the same process.
- ASP.NET, the replacement for ASP, offers compiled web pages (making processing of web requests much more efficient) and includes a large number of pre-written components that can generate commonly used HTML form and user-interface items for you (the so-called *server controls*).
- The main programming languages have been moved far closer together, so code written in VB, C++ and C# can be intermixed. Freely step between the languages in the debugger.
- Components are wrapped up in a new unit called an *assembly*, which is highly self-describing, making installation and use of components very easy.

2.12.2 JSP (JavaServer Pages)

JSP is a web-scripting technology that means for creating dynamic Web-based content using server-side (middle-tier) processing. JSP simplifies the process of creating these dynamic pages by separating the application logic from the page design and encapsulating logic in portable, reusable Java components. Besides it also simplifies the task of building web applications that work with a wide variety of web servers, application servers, browsers and development tools. [8]

JSP technology has evolved from the powerful servlet technology. (Servlets are Java technology-based, server-side applications). (Hall, 2000) JSP extends the servlet technology in many ways, making it easier and faster to build, deploy, and maintain server-side applications that communicate with Web-based clients.

JSP technology builds on the strength of the Java family and the multivendor Java community, extending the core capabilities of the Java platform to create powerful, flexible, and easy-to-maintain dynamic Web pages. JSP technology inherits all of the benefits of the Java language, including platform- and server-independence, a modular and reusable component architecture, and access to the rich family of Java APIs (including JDBC, JavaMail, and Java Transaction Service).

JSP technology can meet the needs for up-to-date, high-performance, and reliable Web-based applications, while addressing the overall organizational or developer's needs for long-term architectural decisions:

Table 2.3: Benefits of JSP

- **Vendor Independence**

As part of the Java technology family, JSP enables rapid development of web-based applications that are platform-independent. It built with the input and support of the major vendors in the market that does not lock companies into any one vendor's solutions.

- **Portability**

JSP technology is portable across platforms and servers alike, building on the Write Once, Run Anywhere philosophy of the Java language. Users can access Java technology-based components (beans, customized JSP tags) that are also reusable and portable across platforms.

- **Flexibility**

The multi tier architecture using JSP technology is inherently flexible and adaptable. Developers can change data sources, access external data, and implement new security or authorization methodology - all without affecting "customer-facing" applications.

- **Low Cost of Ownership**

JSP technology lets page authors handle the ongoing maintenance of dynamic pages without requiring developers to be involved in changes that do not involve application logic.

- **Scalability**

JSP pages are compiled once when first invoked and remain in memory; this provides better scalability for high-volume sites than the others approach.

Therefore JSP is a better technology provides an easy way to create dynamic web pages and simplify the task of building web applications that work with a wide variety of web servers, application servers, browsers and development tools. (Karl et al, 2000)

2.13 DATABASES BEING CONSIDERED

2.13.1 Microsoft Access

The Microsoft access provides one of the easiest ways to create a database. It is easy to be obtained since it is include as part of the Microsoft Office package. Access can be divided in to two different modules to suit the need of different type of users. The first is an easy to use menu driven interface that let user issue commands without an in depth understanding of Access. Program mode lets the user to stored instruction in a file such as Visual Basic file and executes them with one command.

Access allow user to indicate how tables should be related to each other. A table that has referential integrity allows only one parent record for each child record. User can add, delete, and rearrange fields in the table structure. User can also control how the data will be entered in a table using the properties sheet of a field. (Sellappan, 1999)

As a desktop database packages, Access is not design to compete with those system such as Oracle or SQL Server whose engines are superior in terms of speed and multi-user capabilities. Although Access does not provide a good performance in a large user's environment and has been known only able to support a handful of users using it at once, it does has a good performance with limited multi-user capabilities. More over, Access is an ideal front-end application for other database management system such as Oracle and SQL Server and it also integrated well with data transfer between Access and other Microsoft Office application.

2.13.2 MS SQL Server 2000

Today, SQL Server 2000 has been a common name in the rapid development of new generation of enterprise-class business applications that can give company a critical competitive advantage. It provides agility in data management and analysis, allowing organization to adapt quickly and gracefully to derive competitive advantage in a fast-changing environment.

SQL Server 2000 has won many important benchmark awards for scalability and speed. It is a fully Web-enabled database product that provides core support for

Extensible Markup Language (XML) and the ability to query across the Internet and beyond the firewall.

The advantages of SQL Server 2000 are:

i) Fully Web-Enabled

SQL Server 2000 provides extensive database programming capabilities built on Web standards. Rich XML and Internet standard support provide the ability to store and retrieve data in XML format easily with built-in stored procedures. Besides user can also use XML update programs to insert, update and delete data easily.

- **Easy access to data through the Web.** With SQL Server 2000, HTTP can be used to send queries to the database, perform full-text search on documents stored in database, and run queries over the Web with natural language.
- **Powerful, flexible Web-based analysis.** SQL Server 2000 Analysis Services capabilities are extended to the Internet. User can access and manipulate cube data by means of a Web browser.

ii) Highly Scalable and Reliable

With scale up and scale out capabilities, SQL Server meets the needs of demanding e-commerce and enterprise applications.

- **Scale up.** SQL Server 2000 takes advantage of symmetrical multiprocessor (SMP) systems. SQL Server Enterprise Edition can use up to 32 processors and 64 GB of RAM.
- **Scale out.** Scale out distributes the database and data load across servers.

- **Availability.** SQL Server 2000 achieves maximum availability through enhanced fail over clustering, log shipping, and new backup strategies.

iii) Fastest Time-to-Market

SQL Server 2000 is the data management and analysis backbone of the Microsoft .NET Enterprise Servers. SQL Server 2000 includes tools to speed development from concept to final delivery.

- **Integrated and extensible analysis services.** With SQL Server 2000, you can build end-to-end analysis solutions with integrated tools to create value from data. Additionally, you can automatically drive business processes based on analysis results and flexibly retrieve custom result sets from the most complex calculations.
- **Quick development, debugging, and data transformation.** SQL Server 2000 features the ability to interactively tune and debug queries, quickly move and transform data from any source, and define and use functions as if they were built in to Transact-SQL.
- **Simplified management and tuning.** With SQL Server 2000, it is easy to manage databases centrally alongside all enterprise resources. Stay online while easily moving and copying databases across computers or between instances.

Ms SQL Server is outperformed than MS Access and Informix SQL. This is because it includes a superset the ASNI standard SQL language elements that couldn't be find in MS Access and Informix.

Chapter 3: Methodology

3.1 Introduction

The efficiency of a project is usually heavily dependent on the needs for a good and systematic plan to guide and monitor the development process. The use of software process model depicts the process development into a more understandable form.

System or software development generally takes the form of life cycle. This life cycle is referred as the system development life cycle (SDLC). All system goes through the same generic stages in their lifetime (Sommerville, 2001). The stages are:

- Feasibility study
- Analysis and requirement specification
- Design
- Implementation
- Maintenance

The software process consists of a set of steps that encompass methods, tools, and procedures. These steps are often referring to as software engineering paradigm or software life cycle models. There are many software engineering models:

- Build & fix model
- Waterfall model
- Rapid prototyping model
- Spiral model
- And other

The software development process that the author has chosen to develop CAS is the waterfall model.

3.2 Waterfall model

The Waterfall Model is the earliest method of structured system development derived from other engineering processes. Although it has come under attack in recent years for being too rigid and unrealistic when it comes to quickly meeting customer's needs, the Waterfall Model is still widely used. It is attributed with providing the theoretical basis for other Process Models, because it most closely resembles a "generic" model for software development. [1] Waterfall model is very straight forward way of modelling software development processes. It takes the fundamental processes in software development and represents them as separate processes phase. In principle, each phase has to wait for the previous phase to finish before it can start. The output from the previous phase, normally one or more document, will be passed to the current phase. The phases that are listed below:

- Requirement analysis and definition

  This phase will explore the system's services, constraints and goals and is established by consultation with the system stakeholders. Then it will be define in detail to serve as the system specification.

- System and software design

  The system design process partitions the requirement to either hardware or software design. This phase will deals with designing the system architecture.

Software design involves identifying and describing the fundamental software system abstraction and their relationship.

- Implementation and unit testing

  This is the phase where the coding of the system is being done. It will use the system design from the previous phase as the blue print for the system and realised it as a set of programs or program unit. Unit testing involves verifying that each of the unit developed meet its specification.

- Integration and system testing

  All the program units that are developed in the previous will be integrated and tested as a complete system to ensure that the software requirements have been met.

- Operation and maintenance

  This is normally the longest phase in the life-cycle. The system is installed and put into practical used. Maintenance involves correcting error which were not discovered in early stages of the life cycle, improving the implementation of system units and enhancing the system's service as new requirements are discovered. (Sommerville, 2001)

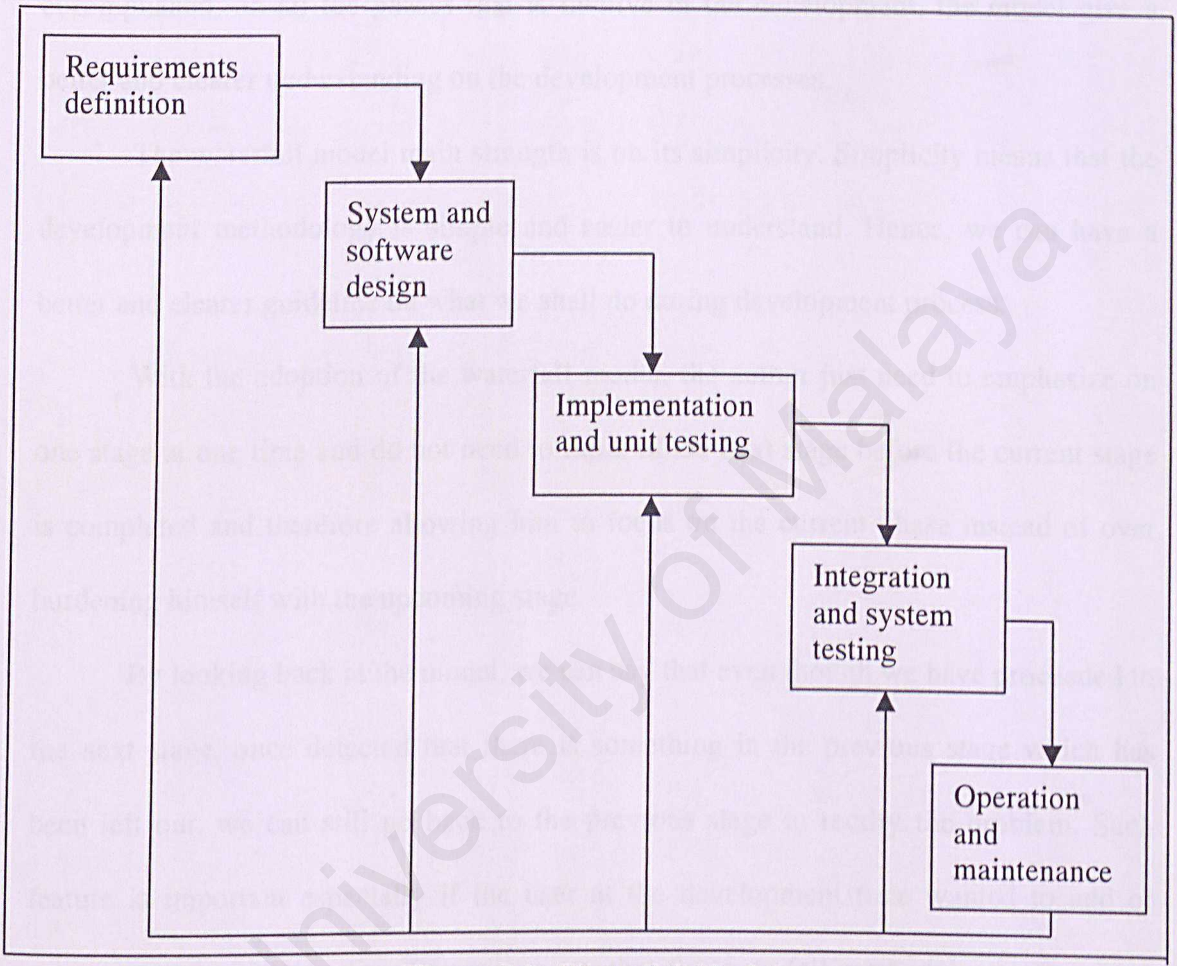

Requirements definition
System and software design
Implementation and unit testing
Integration and system testing
Operation and maintenance

Figure 3.1: Waterfall Model

3.3 Justification of Methodology

The main reason for choosing the waterfall model is because of the simplicity of the model. With a well defined guideline, including the tasks that are needed to be accomplished, on all the phases that is involve in the development, the model give a better and clearer understanding on the development processes.

The waterfall model main strength is on its simplicity. Simplicity means that the development methodology is simple and easier to understand. Hence, we can have a better and clearer guideline on what we shall do during development process.

With the adoption of the waterfall model, the author just need to emphasize on one stage at one time and do not need to think of the next stage before the current stage is completed and therefore allowing him to focus on the current phase instead of over burdening himself with the upcoming stage.

By looking back at the model, we can say that even though we have proceeded to the next stage, once detected that there is something in the previous stage which has been left out, we can still go back to the previous stage to rectify the problem. Such feature is important especially if the user at the development time wanted to add or remove certain functionality. So we can say that the waterfall methodology offers us a backward feature whereby we can rectify errors or incomplete of certain criteria in the previous stage although we have proceeded into the next stage.

The Waterfall model itself can be used for single project that are cost or time restricted because of the easy way to manage it and these restrictions are often an important issue. This methodology emphasizes more on planning rather than rapid development. It allows estimation of the completion of each stage so that the system can be developed within the time frame.

Simplicity is also essential when we are to present or explain or develop the progress to the user especially to those who are not familiar with software development cycle. By now, we can give the user a better understanding on what is going on or what is the progress of the project development. If the author is to adopt a more complex methodology, user may find it difficult to understand what the author trying to tell them

3.4 Technique used to define requirement

The following techniques are used to define the system requirements:

i) Internet surfing

- Internet is being used to seek information on the techniques used in the development of the system such as case based reasoning method

ii) Brainstorming

- Discussion among friends especially those that have some similarity in the method being used, study some information related to the project and have creative thinking can be helpful as providers of information about recent trends and latest technology. Furthermore we can have a research in more specialized sources.

iii) Documents room

- These can be useful sources of information to gain the related information by studies the existing thesis. Several seniors thesis in the FSKTM document room can help to gain the skill of software development such as technology, software architecture and development tools.

iv) Library and Bookshop

- A lot of materials in the library such as journal, conference and reference books offer a relatively concise information and format for research. Meanwhile reference books in bookshop also useful in literature review as it do offer a good starting point intended for teaching and from which to find more detailed sources.

v) Discussion with the supervisor

- A discussion with the supervisor has been practiced to get help and also validate the work being done.

The main techniques being used in developing the CAS is the case based reasoning method. The advantages of using CBR in CAS are given below:

- Using real life case

  Students are normally applying course according to the minimum requirement given in the guideline book. However, in real life, courses minimum requirement is usually much higher than those that have set by the UPU.

- Quicker solutions finding

  By just retrieving and adapting cases in the case base, solution can be get much faster.

- Students can has a much higher chance in applying the courses they wanted

  When a student of certain criteria is accepted in to a course, it is likely that other students of similar criteria will be enrolled in to the course too. Due to this assumption, student will have a higher chance to enrol into those courses.

3.5 Techniques and tools that is used to developed CAS

The main technique that is used in developing is the case based reasoning method. The reason behind this choice is that there are no precise guidelines that can guarantee STPM student with certain result can enter the course that they had applied. Besides that, the success of some educational organization in helping students getting advice on courses that they should applied for have been an important motivator in this choice. These organization will conduct survey on past students, analyse the data, and use those data to help in giving advice to students, much like they way CBR reason with.

The programming language that is chosen to develop CAS is ASP.Net. As ASP.Net is widely used, the reference for it is easily found. Besides, it is also user-friendly making it ideal for me to implement.

The database that has been chosen in developing this system is Microsoft Access. This DBMS has been chosen because of its simplicity in using and availability. Besides that, it is also widely supported.

Chapter 4: System Analysis

4.1 Objectives

In order to get an overview of the system requirements, an extensive analysis is needed. System analysis is an important phase in system development where the necessity of the system functionality is defined here. It is conducted with the following objectives in mind:

i) To ascertain the functional and non-functional requirements of the system.

ii) To determine the hardware and software that is needed in the course of developing the system.

4.2 Requirement Specification

A software specification definition is an abstract description of the services, which the system should provide, and the constraints under which the system must operate. There are two types of requirement analysis, functional requirement and non-functional requirement.

4.3 Functional requirement

Functional requirement is a statement of the services or functions that a system should provide, how the system reacts to particular inputs and how the system should behave in particular situations.

CAS consists of two main modules. The first one is the general module and the second is the advisory module. For the general module, the functional requirement includes the information module and also hyperlink to the relevant website. For the advisory module, the functional requirement consists of 6 modules which are the identify module, matching module, adaptation module, evaluation module, display module and retain module. The figure 4.1 shows the system structure for CAS.

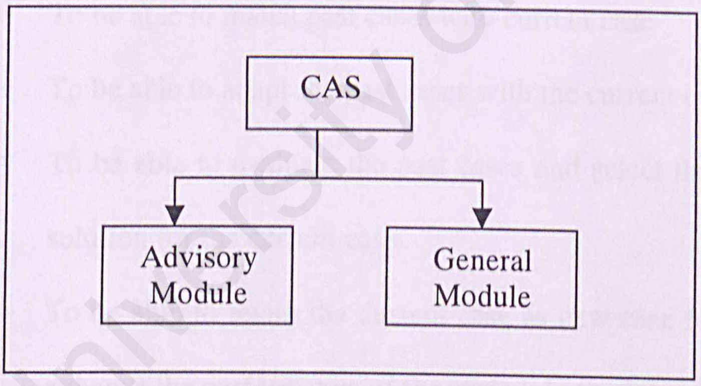


Figure 4.1: System Structure for CAS

4.3.1 The functional requirements for the general module

i) To provide information on the courses

This module enable user to search for the synopsis on the courses offered by University Malaya. This will help the user gain a better understanding on the courses they interested in.

ii) To provide hyperlink to relevant website

This module enables the user to access other relevant website. This will enhance the advisory service of CAS by also providing other sources of information to the user. The sites included are the University Malaya website and also the UPU website.

4.3.2 The functional requirements for advisory module.

i) To implement the CBR cycle

One of the main functional requirements for CAS is to implement the CBR cycle. These include:

- To be able to retrieve cases from the case base
- To be able to match past cases with current case
- To be able to adapt the past cases with the current case
- To be able to evaluate the past cases and select the best cases as the solution for the current case
- To be able to retain the current case as new case for the case base to enhance the performance of the system in the future

ii) To provide information on the courses being advice to apply

As the system display the courses that the user are advice to apply for, the system should also be able to provide information on the courses in order to help the user in making decision for the courses they should applied. The information will help the user gain a better understanding on the courses so that they can make a better judgement of either accepts or decline the advice given by the system.

4.4 Non-functional requirement

Non-functional requirement are essential definitions of the system properties and constraint under a system must operate.

i) User-friendliness

This system is required to have a user-friendly interface to ease user in using the system. This includes usage of menu, descriptive captions etc. to guide users thorough the usage of the system.

ii) Reliability

A system is said to have reliability if it does not produce dangerous or costly failures when it is used in a reasonable manner, that is, in a manner that a typical user expects is normal. This definition recognizes that a system may not always be used in the ways that the designer expects.

iii) Maintainability

The system is also required to have the ability to be maintained and expanded for future enhancement. Therefore, this system is to be developed using common languages such as VB .Net where users can get to learn the language easily or even get some reference point from other people who can provide them with relevant information.

iv) Efficiency

Efficiency in computer technology means a process or procedure that can be called or accepted in an unlimited number of times to produce similar outcomes output at a creditable pace or speeds.

v) Expandability

The degree to which architectural, data, or procedural design can be extended. This means that the system can be able to be expandable in the future.

vi) Modularity

Modularity is an important factor to a good programming design. The working of the system was broken down in to modules so that distinct functions of objects could be isolated from one another. This characteristic makes testing and maintenance going to be much easier.

4.5 Hardware and Software Requirements

Listed here are some of the hardware and software that is being used in developing CAS.

Hardware Requirement:

i) Personal Computer

ii) Pentium IV 1.8 MHz processor

iii) 256 MB RAM

Software:

i) Microsoft Visual Studio .Net

ii) Microsoft Access

Chapter 5: System Design

5.1 Introduction

System design is a very important phase in system development as it determines the success of a system. The system specification describes the features of a system, the components or elements of a system and their appearance to users. Requirements that are found in analysis stage are the one actually translated in to design specification.

The objectives of system design are listed below:

a) Specify logical design element

- Detailed design specifications that describe the features of an information system: input, output, and procedures.

b) Meet user requirements

- Meets user needs stated in term of:

  i) Performing appropriate procedures correctly

  ii) Presenting proper form of information

  iii) Providing accurate results

  iv) Using appropriate method of interaction

  v) Providing overall reliability

c) Easy to use

- Favourable human engineering
- Ergonomic design that is physically comfortable and contributes to user effectiveness and efficiency

d) Provide software specification

- Specific components and functions with adequate detail to construct application software

5.2 System Architecture

CAS uses two main modules to fulfil its functional requirements. The general module is enables user to search for the synopsis of the courses and also provided link to the relevant website, while the advisory module function is to generate an advice for the user. Brief system architecture of CAS is shown below:

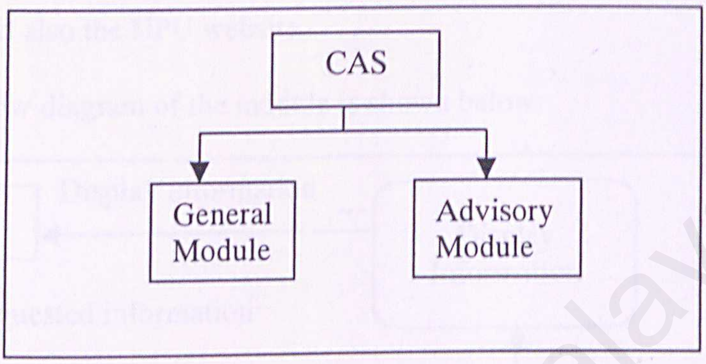


Figure 5.1: System Architecture of CAS

As shown in context diagram, there are 3 external agents involve in CAS. The user who is responsible of entering the problem descriptions and also getting advice from the system, case base which contains previous solves cases and also the course information database which contains the synopsis for the courses.

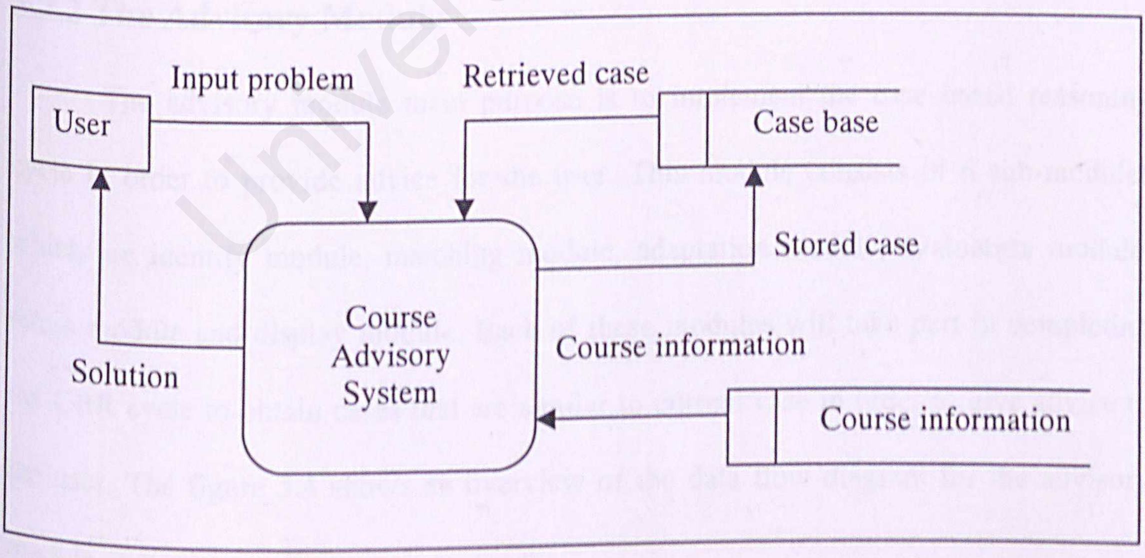


Figure 5.2: Context Diagram for CAS

5.2.1 The General Module

This module enables user to search for the synopsis on the courses and also provide hyperlink to relevant websites. The user will input the identity number of the course and the system will retrieve it synopsis back to the user. Besides that, the user can click on the hyperlink provided to access other relevant websites such as the University Malaya website and also the UPU website.

The data flow diagram of the module is shown below:

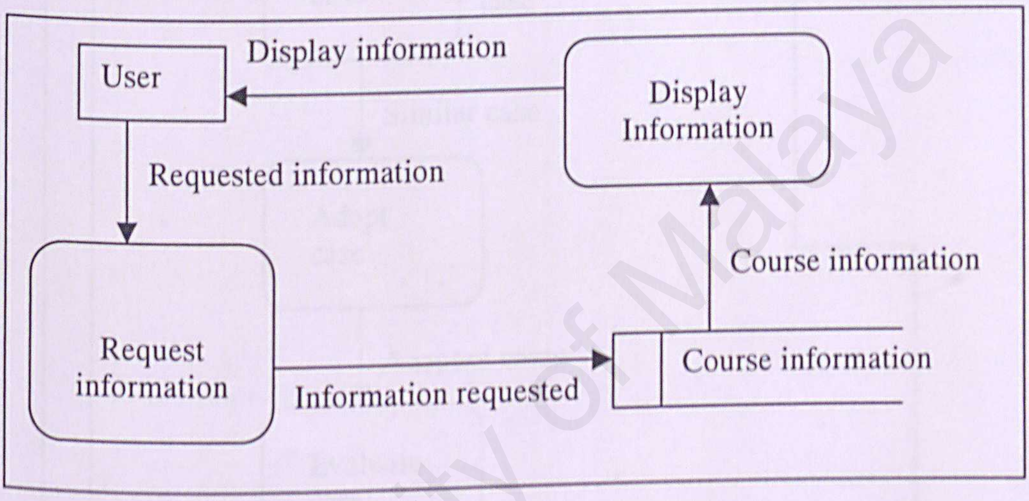


Figure 5.3: DFD for General Module

5.2.2 The Advisory Module

The advisory module main purpose is to implement the case based reasoning cycle in order to provide advice for the user. This module consists of 6 sub-modules which are identify module, matching module, adaptation module, evaluation module, retain module and display module. Each of these modules will take part in completing the CBR cycle to obtain cases that are similar to current case in order to give advice to the user. The figure 5.4 shows an overview of the data flow diagram for the advisory module.

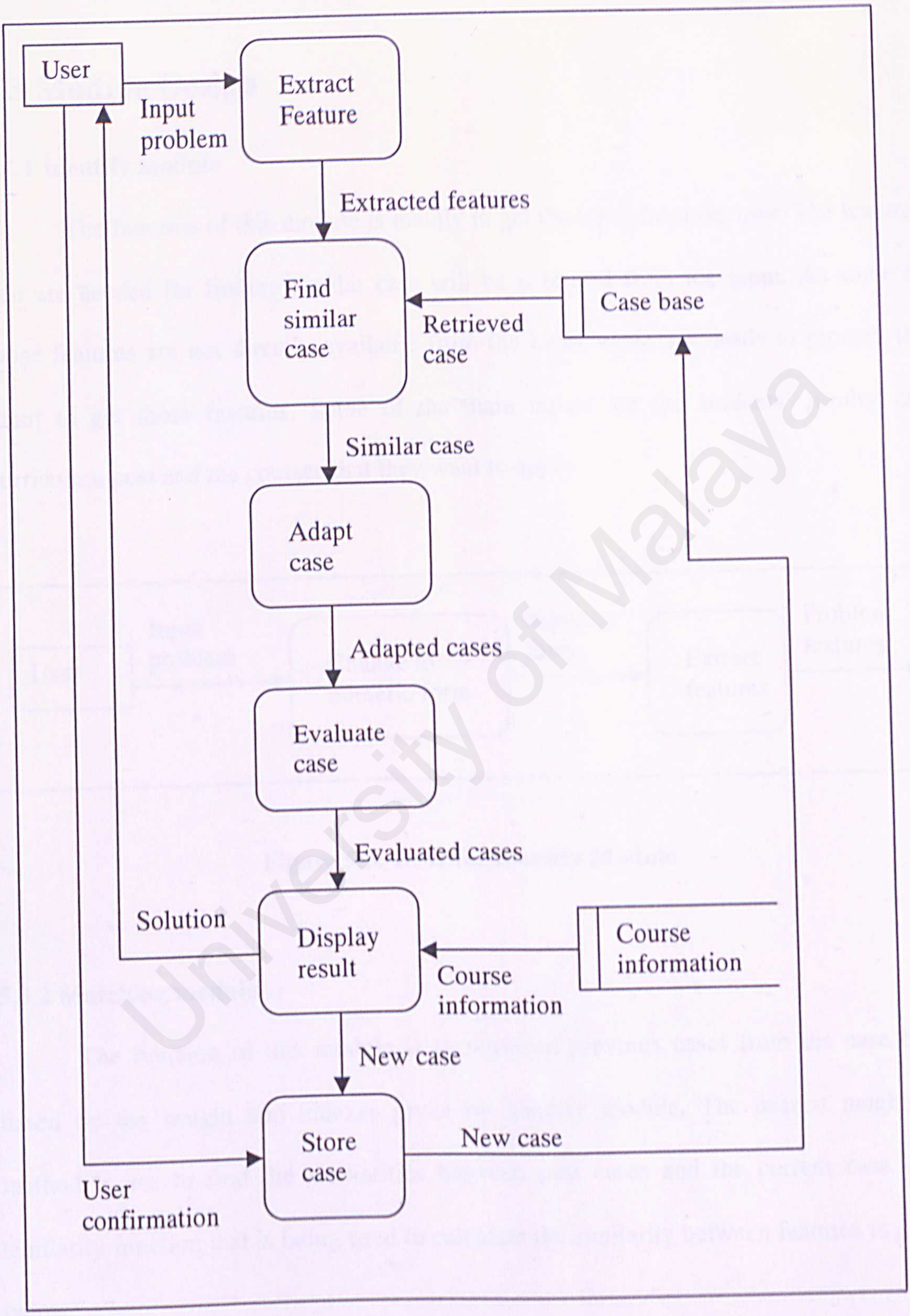


Figure 5.4: DFD for Advisory Module

5.3 Module Design

5.3.1 Identify module

The function of this module is mainly to get the input from the user. The features that are needed for finding similar case will be extracted from the input. As some of those features are not directly available from the input, there are needs to process the input to get those features. Some of the main inputs are the students' results, co-curriculum post and the courses that they want to apply.

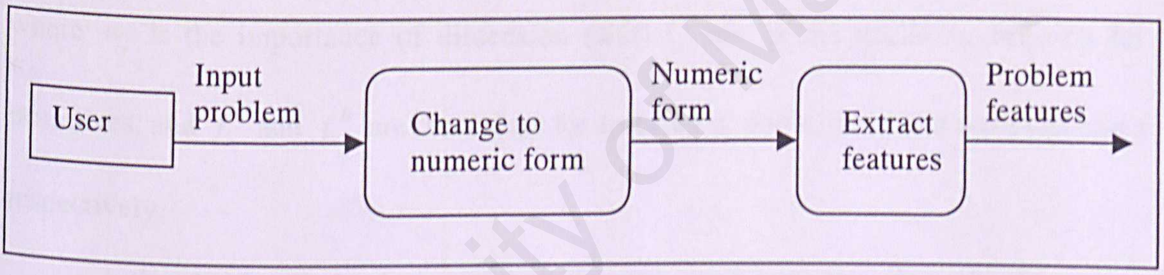


Figure 5.5: DFD for Identify Module

5.3.2 Matching module

The function of this module is to retrieved previous cases from the case base based on the weight and indexes given by identify module. The nearest neighbour method is use to find the similarities between past cases and the current case. The similarity function that is being used to calculate the similarity between features is given below:

$$Sim(I,R) = \frac{Min(I,R)}{Max(I,R)}$$

where I is the input case value and R is the retrieve case value. This equation shows that the minimum value from the both value is divided by the maximum value to find the similarity value between the two features.

Another equation that is use in the matching module is the nearest neighbour equation as given below:

$$\sum_{i=1}^{n} \frac{w_i \times sim(f_i^I, f_i^R)}{\sum_{i=1}^{n} w_i}$$

where w_i is the importance of dimension (slot) I, *sim* is the similarity function for primitives, and f_i^I and f_i^R are the values fcr features f_i in the input and retrieved cases respectively.

A threshold will also been conducted in this module so that cases that have very low overall similarity value will be excluded from consideration. A total of 10 cases will be allowed to pass the threshold value.

The data flow diagram of the matching module is shown below:

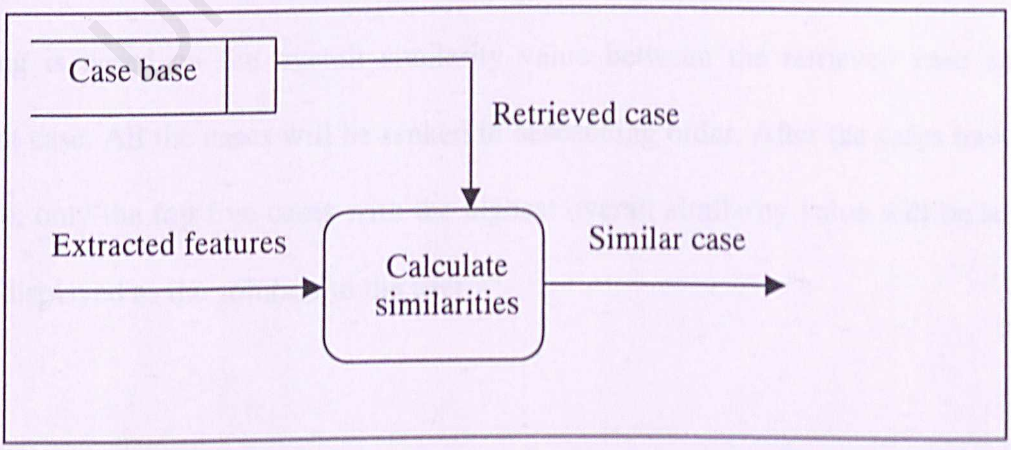


Figure 5.6: DFD for Matching Module

5.3.3 Adaptation module

The function of this module is to adapt the retrieved cases with the current case in order to find the optimum solution for the current case. The adaptation is done base on the courses that the user would like to apply for. Those cases that match with this feature will be added some similarity value to their overall similarity value, thus adapting the solution closer to the user interest.

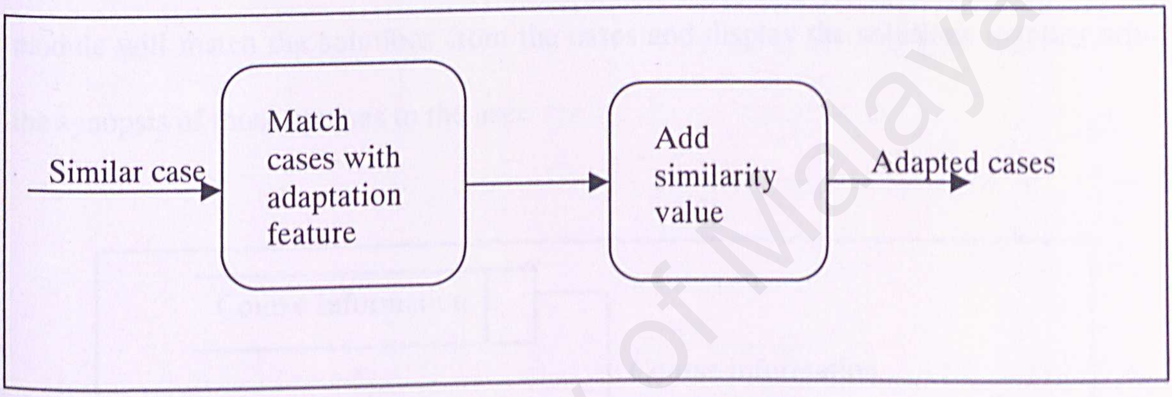


Figure 5.7: DFD for Adaptation Module

5.3.4 Evaluation module

After all the cases have been adapted, it will be ranked by using this module. The ranking is based on the overall similarity value between the retrieved case and the current case. All the cases will be ranked in descending order. After the cases have been ranked, only the top five cases with the highest overall similarity value will be selected to be displayed as the solution to the user.

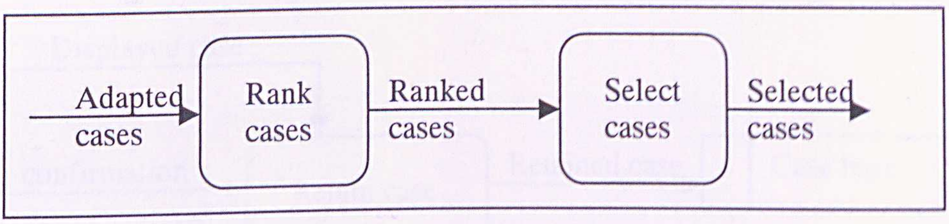


Figure 5.8: DFD Diagram for Evaluation Module

5.3.5 Display module

The display module functionality is to display the result after the CBR cycle. This module will match the solutions from the cases and display the solutions together with the synopsis of those courses to the user.

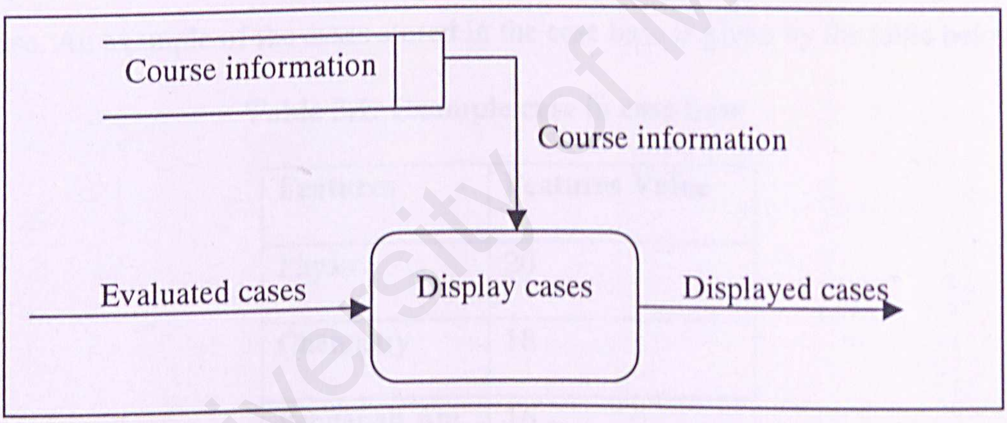


Figure 5.9: DFD for Display Module

5.3.6 Retain Module

The function of this module is to index the new case that the user want to store and stored the case in to the case base so that it can be used later as solution for others problem.

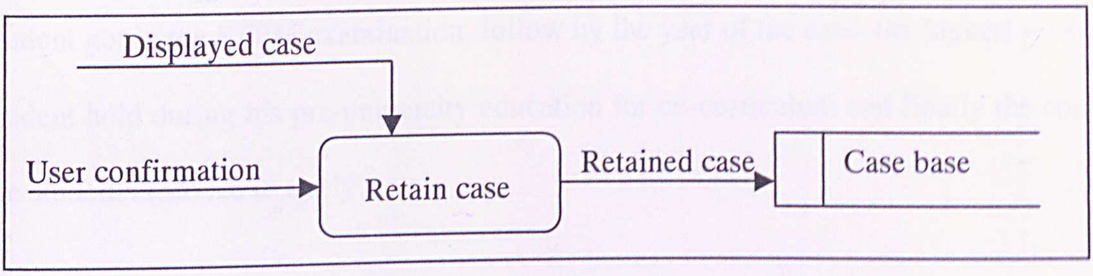


Figure 5.10: DFD for Retain Module

5.4 Case base design

The case base of CAS is built using the Microsoft Access database management system. Although it is develop using a DBMS, but the functionality of the case base is different from database where there is only a single table create to store the cases in the case base. An example of the cases stored in the case base is given by the table below:

Table 5.1: Example case in case base

Features	Features Value
Physic	20
Chemistry	18
Pengajian Am	16
Mathematic	20
Year	2002
Co-Curriculum	President
Course advice	Computer Science

This table shows the features along with its value. This example is made based on the assumption that it is a physic class student. The top four features are the result the student got in the STPM examination, follow by the year of the case, the highest post the student hold during his pre-university education for co-curriculum and finally the course the student is advice to apply.

5.5 Database design

The CAS utilizes the use of a database to support the advisory module by storing the synopsis of the courses in it. The database is called Course Information. This database is used in both the advisory module as well as in the general module. The database is created using the Microsoft Access database management system. The table 5.2 shows the design of the database:

Table 5.2: Database Design

Field Name	**Data Type**	**Size**	**Description**
Course_id	Text	10	An id in the UPU form to identify the courses
Course_des	Memo	120	A description on the courses

5.6 User input design

The user input is an important factor in finding the suitable cases to be used as advice for the user. The table below shows an example of input by the user:

Table 5.3: Example user inputs

User Input Features	Input Value
Physic	A
Chemistry	B

Pengajian Am	C
Mathematic	A
Student Category	Science-Physic
Course hope to apply	Science Computer

This table shows the input of a student with the assumption that the student is a science-physic student. The top four features are the student results which will be converted in to numerical form later, follow by the student category, student interest and course the student would like to apply. The last two features will be used in adaptation of the past cases to the current problem.

Chapter 6: SYSTEM IMPLEMENTATION

6.1 Introduction

System implementation is a process that converts the system requirements and design into program codes. It is the process takes place after the system design phase. This phase describes how the initial and revised process design put into the real work. Besides that, this phase also going to discuss about the coding methods used during the development of CAS.

6.2 Development Environment

The development environment is crucial for the rapid development of CAS. The hardware and software tools used to develop and documents the entire system is as discussed as below:

Hardware Requirements

The following hardware specifications have been used to develop CAS:

a) 1.8 GHz Pentium IV

b) 256 MB RAM

c) 52X CD-ROM

d) 40 GB Hard Disk

e) Other Standard desktop PC components

Software Requirements

i) Tools Used For System Design

The software tools used for system development are vital to successful implementation of this system. Table 6.1 below lists all the software used to develop the system.

Table 6.1: Software used in developing CAS

Software	Usage	Description
Microsoft Windows XP Professional Edition	System Requirement	Operating System
Microsoft Access	System Requirement	Database
Internet Information Service	System Requirement	Web server service. Map local directory to virtual directory and create local web site.
Microsoft Visual Studio .Net Enterprise Architect 2003	System Development	Web Page Coding
Internet Explorer 6.0	System Development	Web Page Browsing
Adobe Photoshop 7.0	System Development	Graphical Design

ii) Software Tools for Report Writing

Microsoft Office Word XP is used to write the report for this system.

iii) Software Tools for Database

The database for CAS is used as a case base library in CBR. It is created using the Microsoft Access.

6.3 Program Coding

Methodology

CAS is developed using a modular approach where each module is developed separately and are later integrated into a fully functional system. For each module, it is further refined into functions and procedure. By using a modular approach, future modification and enhancements are made easily.

Web Page Development

Being a fully web-based advising application, CAS makes use of the internet browser. It has been coded by using the hypertext markup language (HTML).

Besides that, ASP .Net technology is being used. ASP .Net eases database retrieval and manipulation. The web server processes ASP .Net files before being presented to the browser. The user will not be able to view the code written in ASP.

All ASP .Net code will be interpreted into HTML codes, by the web server whenever users request the ASP .Net files. Microsoft Visual Studio .Net 2003 Enterprise

Edition is used to develop the web pages and debugging the errors. They provide a lot of function and wizard in helping development process.

6.4 Case Based Reasoning Implementation

The implementation of CBR on CAS is using the database as the case base to store the cases. Besides that, there is some modification on the CBR cycle where the retaining cycle has been left out so that the integrity of the cases in the case base can be maintained. The features that have been identified to be used to develop the cases are analysis from the university in-take application form. Those features are as below:

i) The four subject students take during STPM

ii) The year of STPM take

The result of the students will be change in to numeric form in order to compare with those in the case base. Table 6.2 shows the conversion:

Table 6.2: Conversion of the result to numeric form

Result	Numeric Value
A	120
B	100
C	80
D	60
E	40
F	20

Besides that, the weight that is being used in CBR is shown below:

Features	Weight
Subject	30
Year	10

With these values, the CBR is implemented as a function as shows in figure 6.1:

```
Public Function CalculateSimilarity(ByVal dSet As DataSet)
As DataTable
        'ADD IF STATEMENT,IF NO ROW RETURN TELL USER USE
OTHER BUTTON ELSE FOLLOW THIS

        Dim dv As New DataView(dSet.Tables("tblTemp"))

        With dv.Table.Columns.Add("sub1", GetType(Decimal))
            .Expression = "iif(subject1 >= " &
dl1.SelectedValue & " , " & dl1.SelectedValue & "/subject1,
subject1/" & dl1.SelectedValue & ")"
        End With

        With dv.Table.Columns.Add("sub2", GetType(Decimal))
            .Expression = "iif(subject2 >= " &
dl2.SelectedValue & " , " & dl2.SelectedValue & " /subject2,
subject2/" & dl2.SelectedValue & " )"
        End With

        With dv.Table.Columns.Add("sub3", GetType(Decimal))
            .Expression = "iif(subject3 >= " &
dl3.SelectedValue & "  , " & dl3.SelectedValue & "
/subject3, subject3/" & dl3.SelectedValue & " )"
        End With

        With dv.Table.Columns.Add("sub4", GetType(Decimal))
            .Expression = "iif(subject4 >= " &
dl4.SelectedValue & "  , " & dl4.SelectedValue & "
/subject4, subject4/" & dl4.SelectedValue & " )"
        End With
```

```
With dv.Table.Columns.Add("Tahun", GetType(Decimal))
            .Expression = "iif(Year >= " & CInt(TextBox5.Text) & " ,
" & CInt(TextBox5.Text) & "/Year, Year/" & CInt(TextBox5.Text) & ")"
        End With

        With dv.Table.Columns.Add("jumlah", GetType(Decimal))
            .Expression = "(sub1*30 + sub2*30 + sub3*30 + sub4*30 +
tahun*10)/130*100"
        End With


        With dv.Table.Columns.Add("adapt", GetType(Decimal))
            .Expression = "iif(Course_Advise = '" &
ddl1.SelectedValue & "' , jumlah + 5, jumlah)"
        End With

        'Sorting of dataview
        dv.Sort = " adapt DESC"

        'Creating dataTable
        Dim dt As New DataTable

        dt.Columns.Add("Course Advise", GetType(String))
        dt.Columns.Add("Similarity", GetType(Decimal))

        Dim i As Integer
        Dim j As Integer


        'Adding the 1st courses with highest similarity to datatable
        Dim dr As DataRow = dt.NewRow()
        dr(0) = dv.Item(0).Row(7)
        dr(1) = dv.Item(0).Row(14)
        dt.Rows.Add(dr)

Dim set1 As Boolean = True

        'Choosing the highest similar cases for each courses from
dataview and insert to datatable
        For i = 1 To dv.Table.Rows.Count - 1
            set1 = True
            For j = 0 To dt.Rows.Count - 1
                If dv.Item(i).Row(7) = dt.Rows(j).Item("Course
Advise") Then
                    set1 = False
                End If
            Next
            If set1 = True Then
                Dim dr1 As DataRow = dt.NewRow()
                dr1(0) = dv.Item(i).Row(7)
                dr1(1) = dv.Item(i).Row(14)
                dt.Rows.Add(dr1)
            End If
        Next
        CalculateSimilarity = dt
    End Function
```

Figure 6.1: Implementation of CBR as a function in CAS

Chapter 7: System Testing

7.1 Introduction

Testing is the process of exercising of evaluation a system by manual or automatic means to verify that it satisfied requirements or to identify differences expected and actual results. Testing probably the least understood part of a software development project. A bug is any unexpected, questionable, or undesired aspect or behavior displayed, facilitated, or caused by the software being tested. Testing can uncover different classes of errors in a minimum amount of time and with a minimum amount of effort. The strategies used for testing are unit testing, integration testing and validation testing and system testing.

7.2 Unit Testing

Unit testing verifies that the component functions properly with the types of input expected from studying the component's design. The first step is to examine the program code by reading through it, trying to spot algorithms, data and syntax faulty. This is followed by comparing the code with specifications and with the design to make sure that all relevant cases have been considered. Next, the browser is used to view the result/web page and then eliminate remaining syntax faulty if necessary. Finally, test cases are being developed to show that the input is properly converted to the desired output.

7.3 Integration Testing

Integration testing is a systematic technique for constructing the program structure while conducting tests to uncover errors associated with interfacing. The objective is to take unit tested modules and build a program structure that has been dictated by design. This testing will ensure that the interfaces such as the module calling sequence in CAS are arranged correctly.

The incremental is the antithesis of the high bang approach. CAS's program is constructed and tested in small segments, where errors are easier to isolate and correct; interface are more likely to be tested completely. Bottom – up approach begins construction and testing with modules at the lowest levels in the system and then moving upward to the modules at the higher levels. Regression Testing is another incremental approach, which is the re-execution of some subset of tests that have already been conducted to ensure that changes have not propagated unintended side effects. It is the activity that helps to ensure that changes (due to testing of for other reasons) do not introduce other behavior or additional errors.

7.4 Validation Testing

At the culmination of integration testing, CAS is completely assembled as a package; interfacing errors have been uncovered and corrected, and a final series of software tests – validation testing – may begin.

Software validation is achieved through a series of black box tests that demonstrate conformity with the requirements. For CAS, a test plan outlines the classes of tests to be conducted, and a test procedure defines specific test cases that will be used in an attempt to uncover errors in conformity with requirements. A total of 20 test cases that are obtained from the educational organization are used to validate the system. The result of the test is shown in figure 7.1:

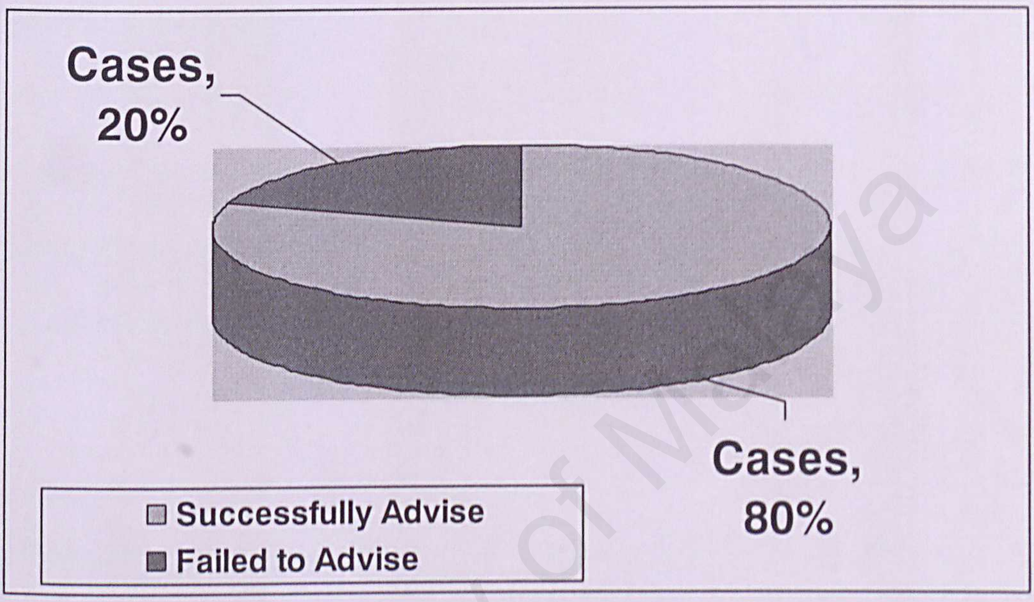


Figure 7.1: Result from the testing phase using real data

A case is considered successful if it has a similarity higher that 90%.

Alpha testing is used to test CAS to ensure that errors that only the end user seems able to find.

7.5 System Testing

The last testing procedure done is system testing. Testing the system is very different from unit testing, integration testing and validation testing. The objective system testing is to ensure that the system does what the users want it to do.

CAS is tested whether it meets specific performance testing. Data integrity testing is used to verify that data is stored in a manner where it is not compromised under updating, restoration or retrieval processing in CAS.

Chapter 8: SYSTEM EVALUATION AND CONCLUSION

8.1 Problems and Solution

As this project has to be completed within a short of time, so during the development of this system, a lot of technical issues needed to be resolved and also a lot of problems had been encounter. Solutions have been sought during the time of development, testing and also made some of the reference to my course mate. As a result, by encountering with these problems has been proven to be valuable learning experience.

8.1.1 Problems and Solutions during System Studies and Analysis

During the studies and analysis phase, a lot of studies have been carried out. Lack of knowledge about the web application and also on case based reasoning has been a great obstruction.

8.1.1.1 Determining Project Scope

As CAS involved developing a web-based course advisory system, to built a full-fledged system is merely impossible within the given time frame. Inexperience with the current technologies and particular scripting language is another hindrance to implement a truly robust advisory system.

A number of discussions were held with project supervisor and also research on current application to university is conducted to outline the scope of the project to be building during the initial stages of the project.

8.1.1.2 Difficulties in Choosing a Programming Language

As we know, there are a lot of programming languages available in the market. A number of them can be used to develop a web application. Due to this reason, it is hard to determine which language is the best approach to be use. So, seeking advises and views from project supervisor and also from some of the course mate whom engaging in similar project are carried out. After much reference, studies and surveys, ASP .Net is chosen as the approach to developed CAS.

8.1.2 Problems and Solutions during System Implementation And Testing

The problem faced during the initial project studies and analysis, were not as crucial compared to the problems faced during implementation and testing period.

8.1.2.1 Lack of Experience in Web-based Programming

As there is no prior knowledge in programming in a web-based environment, a lot of studies need to be done. New programming language like VB .Net, ASP .Net and HTML are needed to be learnt within a short period of time span. Besides, programming concepts for web application is quite different from the traditional way of programming.

However, these obstacles are resolved through discussions with course mates and supervisor, reference books and also from the Internet.

8.2 System Limitation

The limitations for CAS are listed below:

i) Browser Limitation

CAS can only run on the Internet Explorer 4.0 and above

ii) Microsoft .Net Framework

The web server that is being used need to install the Microsoft .Net Framework

iii) Internet Information Service (IIS)

The web server needs to install Internet Information Service component

iv) Cases in case base

The number of cases in the case base should be large and recent enough to give an accurate advice.

v) Selected Combination of subject available

Only a selected combination of subject offered in STPM that are available to be match in the case base.

8.3 Experience Gained

Through the process of developing this system, a lot of useful experiences have been gained. Below describes some of the experiences:

i) By developing the system, some experiences have been gained in developing a web-based application

ii) Able to learn the ASP .Net, which required to develop a web-based application

iii) Get to experience and used the knowledge gained from the database course in creating, inserting, updating and modifying a database by using the SQL statements

iv) Able to learn VB .Net that is being used in ASP .Net

8.4 Future Enhancement

Future enhancements have to been done to make the system more robust, accurate and useful in the future. System development has no boundaries as new requirements and better implementation methods continue to arise and evolve. There are several enhancements that could extend the usability of the developed system.

8.4.1 Extending the involving university

Currently, CAS is only available to be used on courses that are being offered by University of Malaya. In the future, CAS can be adopted to be used on courses offered by other university such as USM, UPM, UUM and UKM. Such availability surely will help students making better judgment in choosing the courses in university.

8.4.2 Extending the combination of subjects

CAS only available to some of these combinations that is more famous among STPM students as there are a lot of combinations of subjects in STPM. It can be upgraded to include other combination of subjects.

8.4.3 Moving to SQL Server

Currently, CAS is using Microsoft Access as the case base as the number of cases that are available is limited. In the future, if there are a large number of cases available, it should be moved to SQL Server which is more reliable and also faster. Besides SQL Server, it can also be implemented in to other database such as MySQL, Oracle and Sybase.

8.4.4 Analyzing Trends

In the future, this system can used to collect data regarding the trends in university in-take application. This can be useful to determine courses that the students would like to enrol.

Conclusion

Overall, this web-based course advisory system has achieved and fulfilled the objectives and requirements as an advising system on courses to apply during university in take.

However, CAS can be further enhanced to become a more powerful and sophisticated advisory system in the future. There are still many rooms for improvement for CAS in terms of implementation of a comprehensive students' advisory system.

There was a lot of knowledge gained throughout the development of this system. The knowledge includes as below:

i) Get more used to Internet environment
ii) Web application development
iii) Internet technologies
iv) Several of web-based programming language such as ASP .Net, HTML, VB .Net
v) Database server and web server
vi) Application of A.I. technique in daily problems
vii) Detail of Case Based Reasoning

Actually, the programming using ASP .Net, HTML and VB .Net proves to be a valuable experience. Then, the interface design using Adobe Photoshop provides a good start into web page design. But while programming skills and techniques are important in developing the system, it must not discount that in any system development, a good software engineering techniques must also be applied.

Finally, all the problems that faced and experienced will be going to be very useful in my future endeavours because the era is now moving towards the Internet technologies.

Appendix A: User Manual

Introduction

Course Advisory System (CAS) is an internet application that consists of 2 main module which are general module (implemented as the Course Information Page) and advisory module (implemented as the Advisory Information Page). CAS enables the users especially STPM students to get advise for courses offered by University of Malaya that they should applied for in their university in-take. Besides that, students can refer to the synopsis on those courses as guidance. It uses the CBR technique to advice students on those courses.

This system is easy to use and learn where all the function in this system is meaningfully descriptive and can be easily executed by a simple point and click on the function button and hyperlink.

This manual will be able to give users a guide to all the function available in the system. This manual includes the following:

i) Course Information Page users guide

ii) Advisory Information Page users guide

Case Based Reasoning

Case Based Reasoning (CBR) is an artificial intelligent technique that tries to mimic the way human reason when solving problem. By using previous solved cases that is store in the case base, the system will tried to match current problem description (the result, etc) with cases in the case base and displayed cases that have a high similarity as the solution.

Requirements for running CAS

The minimum hardware configurations requirements for running CAS system listed as below:

i) A Pentium 3 processor or above

ii) Minimum 64MB RAM

iii) Modem/NIC to connect to internet or Web Server

iv) Keyboard and mouse as input device

CAS requires the following software as its running platform.

i) Windows 98 or newer version

ii) Microsoft Internet Explorer 4.0 and above

Getting Started with CAS

CAS is not like the ordinary system or application that needs installation process before running it. CAS is a web site that provides services to users, meaning that it does away with the need to install a program into hard disk in order to run it. Before accessing to CAS, please make sure that user computer meets the minimum hardware and software requirements stated.

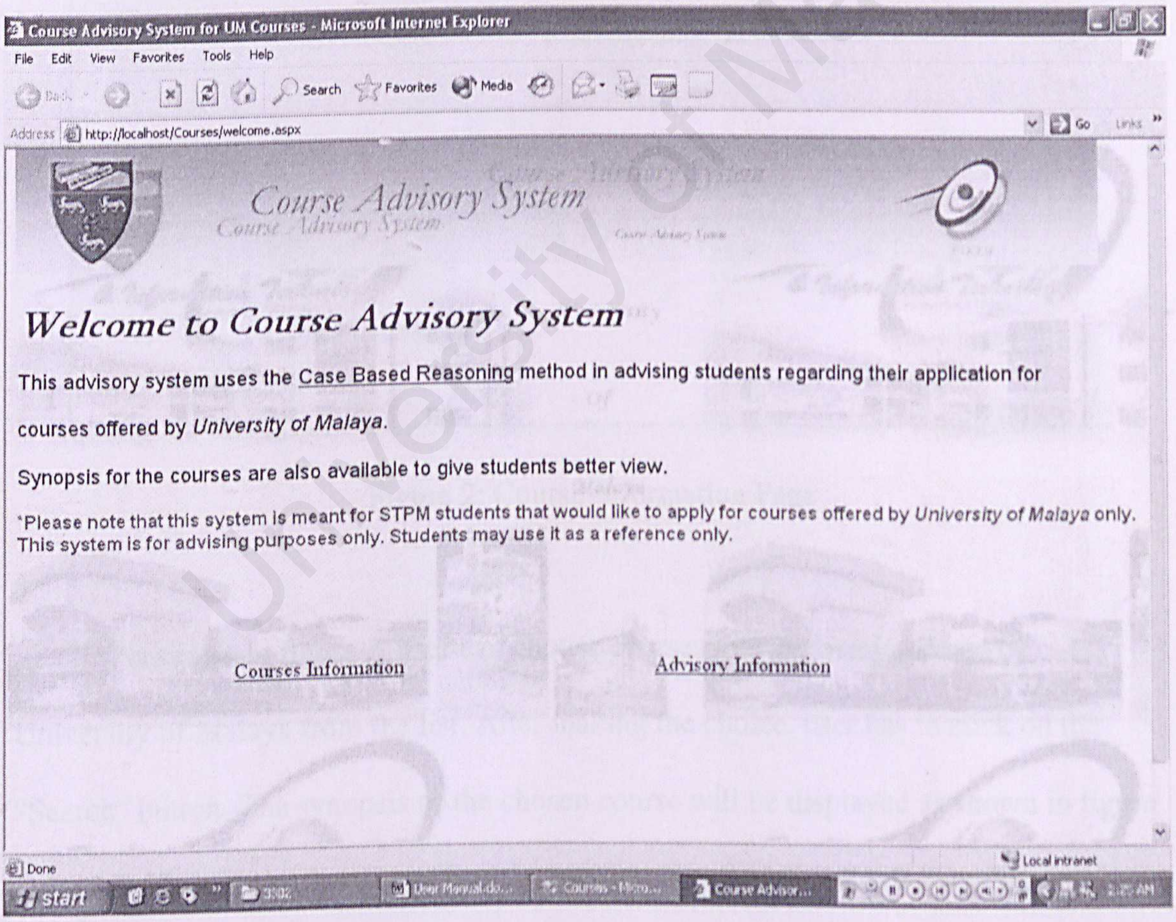
Using CAS

Firstly, user need to start up the Internet Explorer browser by clicking the con, which located on user desktop or user may go to the Start Menu and then find the Internet Explorer option and then select it.

Once user have successfully started the Internet Explorer, user need to type in the following URL into the URL location:

http://CAS/welcome.aspx

As a successful result of browsing to the address above, the homepage of CAS will be shown on user monitor's screen as in Figure 1. Within the page, user may choose whether user would like to look for courses synopsis or use the CBR function to get advice on courses based on user results. A brief explanation on Case Based Reasoning is also provided here by clicking on the hyperlink.



Welcome to Course Advisory System

This advisory system uses the Case Based Reasoning method in advising students regarding their application for courses offered by *University of Malaya*.

Synopsis for the courses are also available to give students better view.

*Please note that this system is meant for STPM students that would like to apply for courses offered by *University of Malaya* only. This system is for advising purposes only. Students may use it as a reference only.

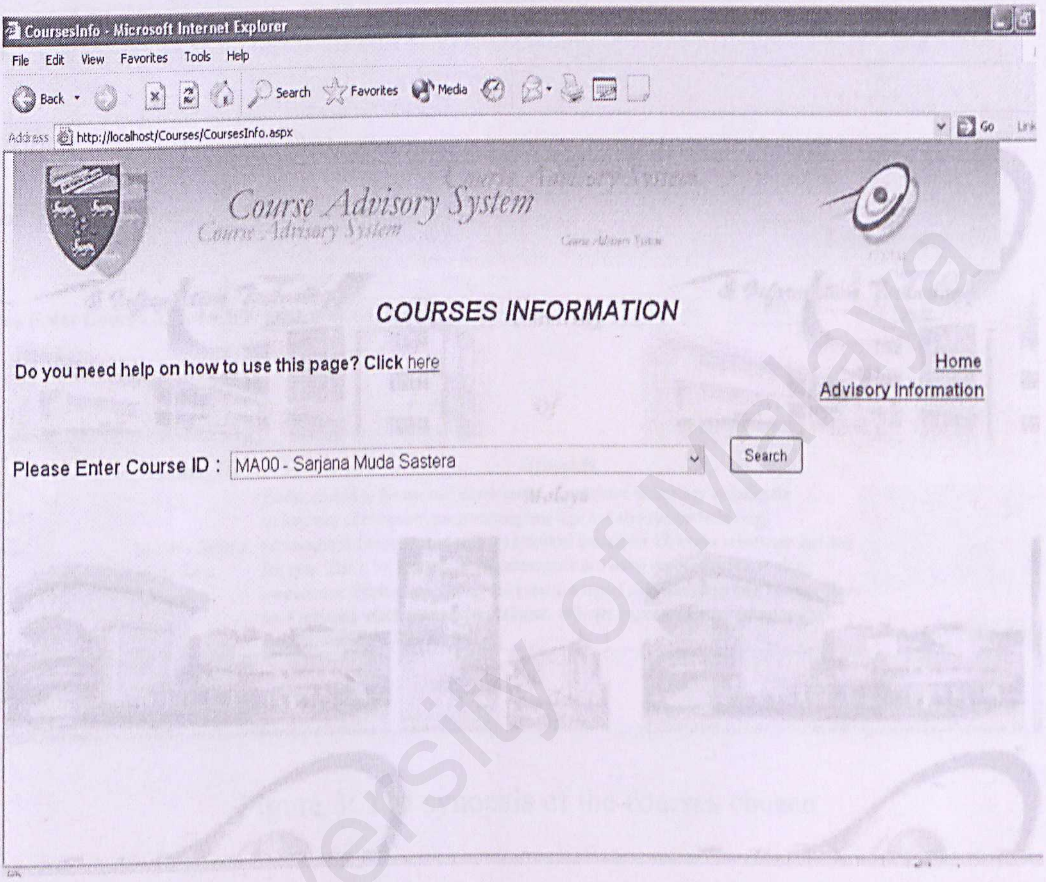
Courses Information

Advisory Information

Figure 1: The main page for CAS

Course Information Page

This page is used to display the courses synopsis based on the courses that user have choose.


CoursesInfo - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back Search Favorites Media
Address http://localhost/Courses/CoursesInfo.aspx
Go
Course Advisory System

COURSES INFORMATION

Do you need help on how to use this page? Click here

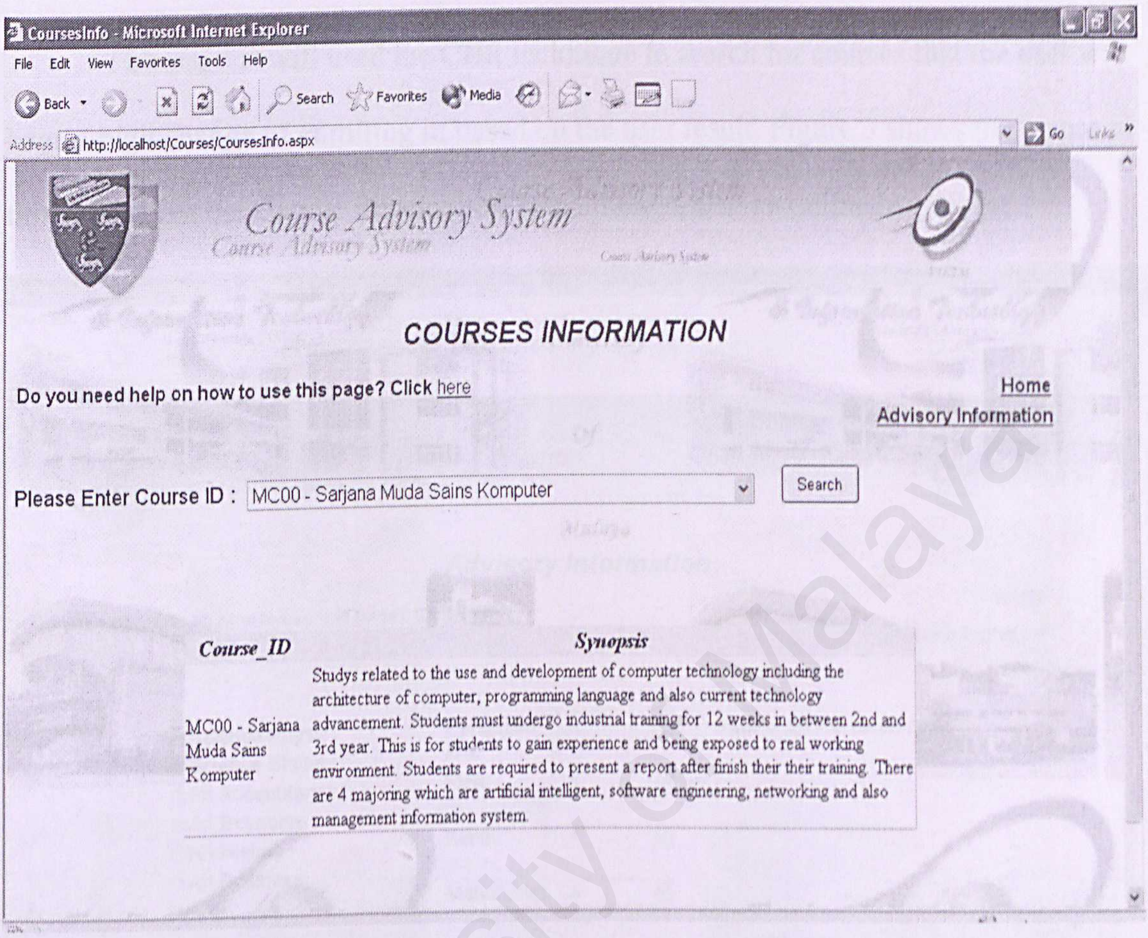
Home
Advisory Information

Please Enter Course ID : MA00 - Sarjana Muda Sastera

Search

Figure 2: Course Information Page

As shows in figure 2, user can choose courses that are being offered by University of Malaya from the list. After making the choice, user has to click on the "Search" button. The synopsis of the chosen course will be displayed as shown in figure 3. Besides that, online help page is also provided for users that didn't know how to use the page. Besides that, user can also access the home page and also the Advisory Information Page from here.


CoursesInfo - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back Search Favorites Media
Address http://localhost/Courses/CoursesInfo.aspx
Go Links
Course Advisory System

COURSES INFORMATION

Do you need help on how to use this page? Click here

Home

Advisory Information

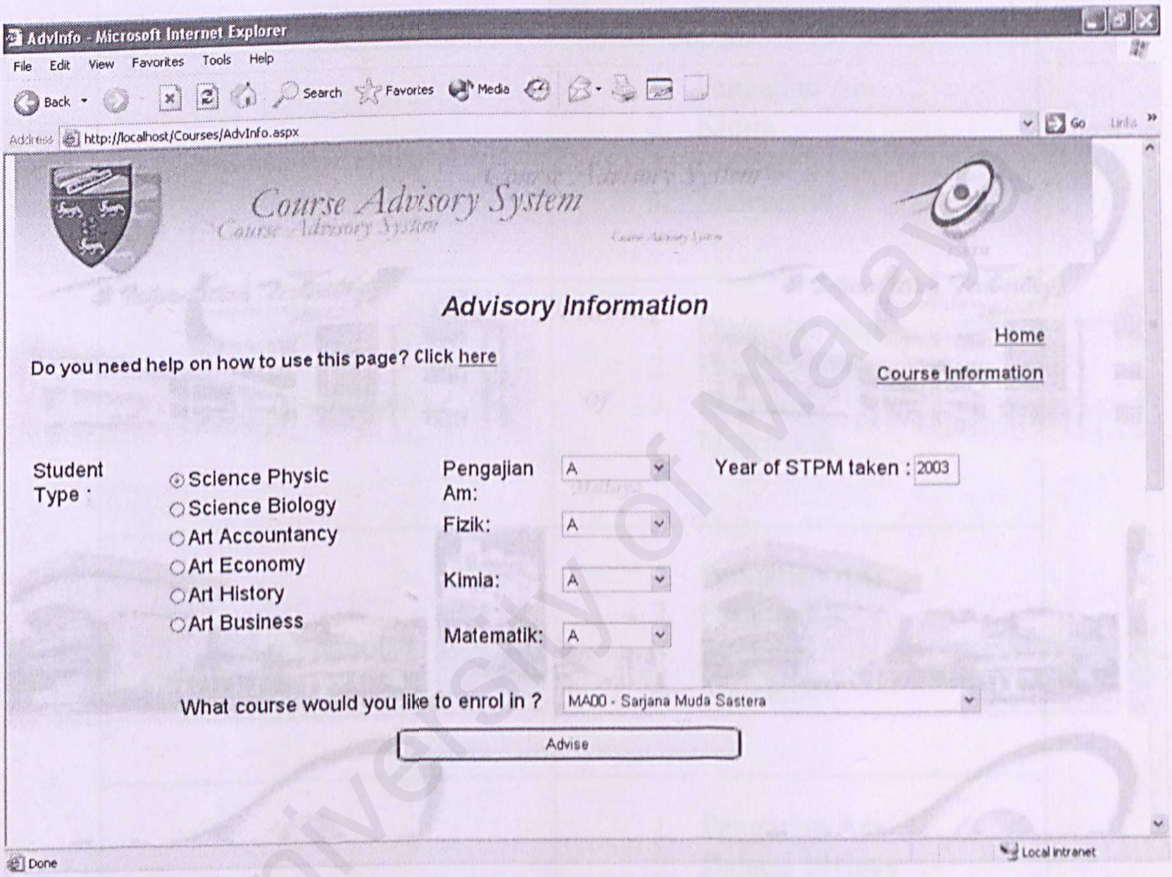
Please Enter Course ID : MC00 - Sarjana Muda Sains Komputer [Search]

Course_ID	Synopsis
MC00 - Sarjana Muda Sains Komputer	Studys related to the use and development of computer technology including the architecture of computer, programming language and also current technology advancement. Students must undergo industrial training for 12 weeks in between 2nd and 3rd year. This is for students to gain experience and being exposed to real working environment. Students are required to present a report after finish their their training. There are 4 majoring which are artificial intelligent, software engineering, networking and also management information system.

Figure 3: The synopsis of the courses chosen

Advisory Information Page

This page is will used the CBR technique to search for courses that the user will have a high change of enrolling in based on the user result. Figure 3 shows the layout of this page. Like the Course Information Page, there is also an online help page.



http://localhost/Courses/AdvInfo.aspx

Advisory Information

Do you need help on how to use this page? Click here

Home

Course Information

Student Type :

● Science Physic
○ Science Biology
○ Art Accountancy
○ Art Economy
○ Art History
○ Art Business

Pengajian Am: A

Fizik: A

Kimia: A

Matematik: A

Year of STPM taken : 2003

What course would you like to enrol in ? MA00 - Sarjana Muda Sastera

Advise

Figure 3: Advisory Information Page

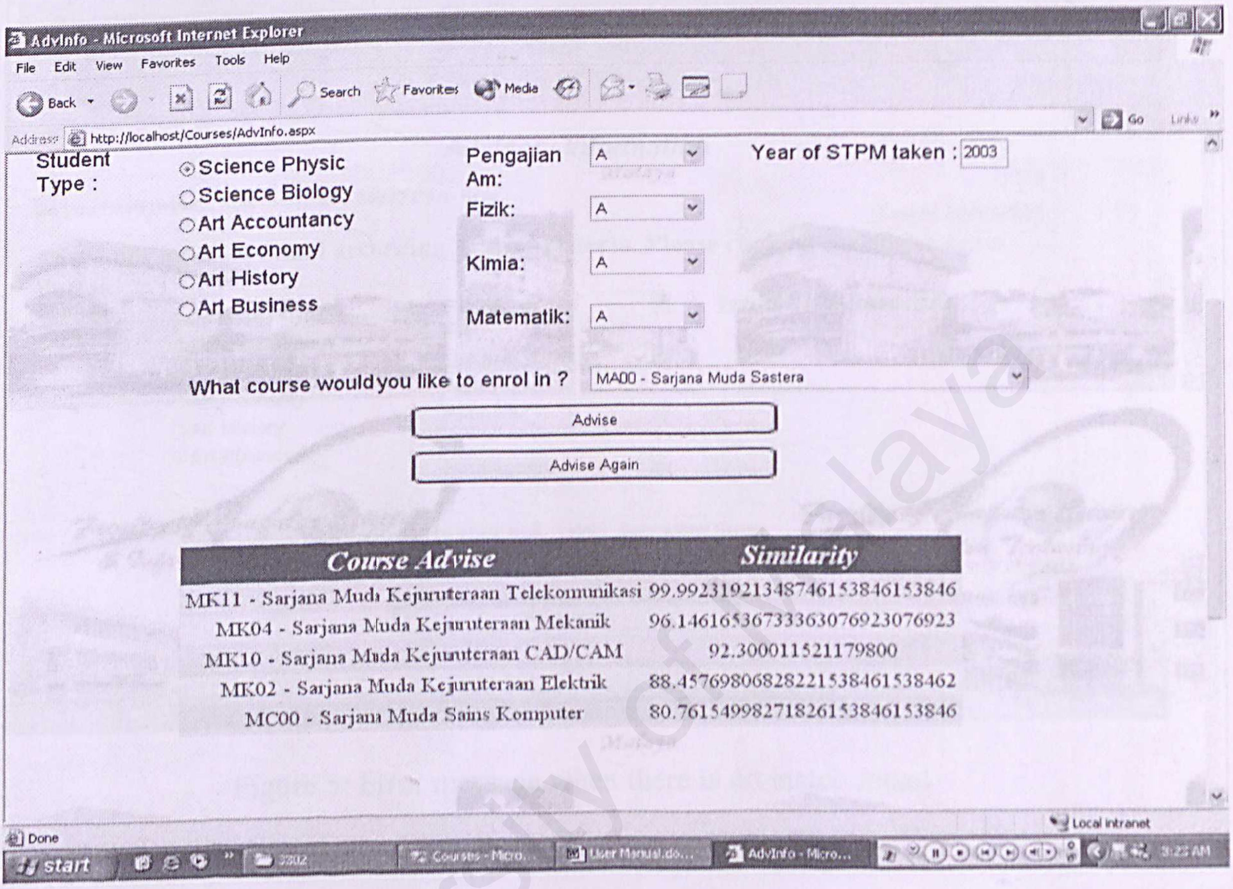
The first step is to choose the student type based on the subject that they took in STPM. The categories of the student type are shows in table 1:

Table 1: The categories of student type

Student Type	Subjects
Science Physic	1. Pengajian Am 2. Kimia 3. Fizik 4. Matematik
Science Biology	1. Pengajian Am 2. Kimia 3. Biologi 4. Matematik
Art Accountancy	1. Pengajian Am 2. Pengakaunan 3. Ekonomi 4. Matematik
Art Economy	1. Pengajian Am 2. Perniagaan 3. Ekonomi 4. Matematik
Art History	1. Pengajian Am 2. Bahasa Melayu 3. Sejarah 4. Ekonomi
Art Business	1. Pengajian Am 2. Bahasa Melayu 3. Sejarah 4. Perniagaan

After the student category have been chosen, user can enter their result on subjects, the year STPM is taken and also the course that user would like to enrol in.

After all data have been entered, click on the "Advise" button. The system will displayed the advice courses together with their similarity as shows in figure 4.


AdvInfo - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back Search Favorites Media
Address http://localhost/Courses/AdvInfo.aspx
Go Links
Advise
Advise Again
Done
Local intranet
start
Courses - Micro...
User Manual.do...
AdvInfo - Micro...

Student Type :

● Science Physic
○ Science Biology
○ Art Accountancy
○ Art Economy
○ Art History
○ Art Business

Pengajian Am: A

Fizik: A

Kimia: A

Matematik: A

Year of STPM taken : 2003

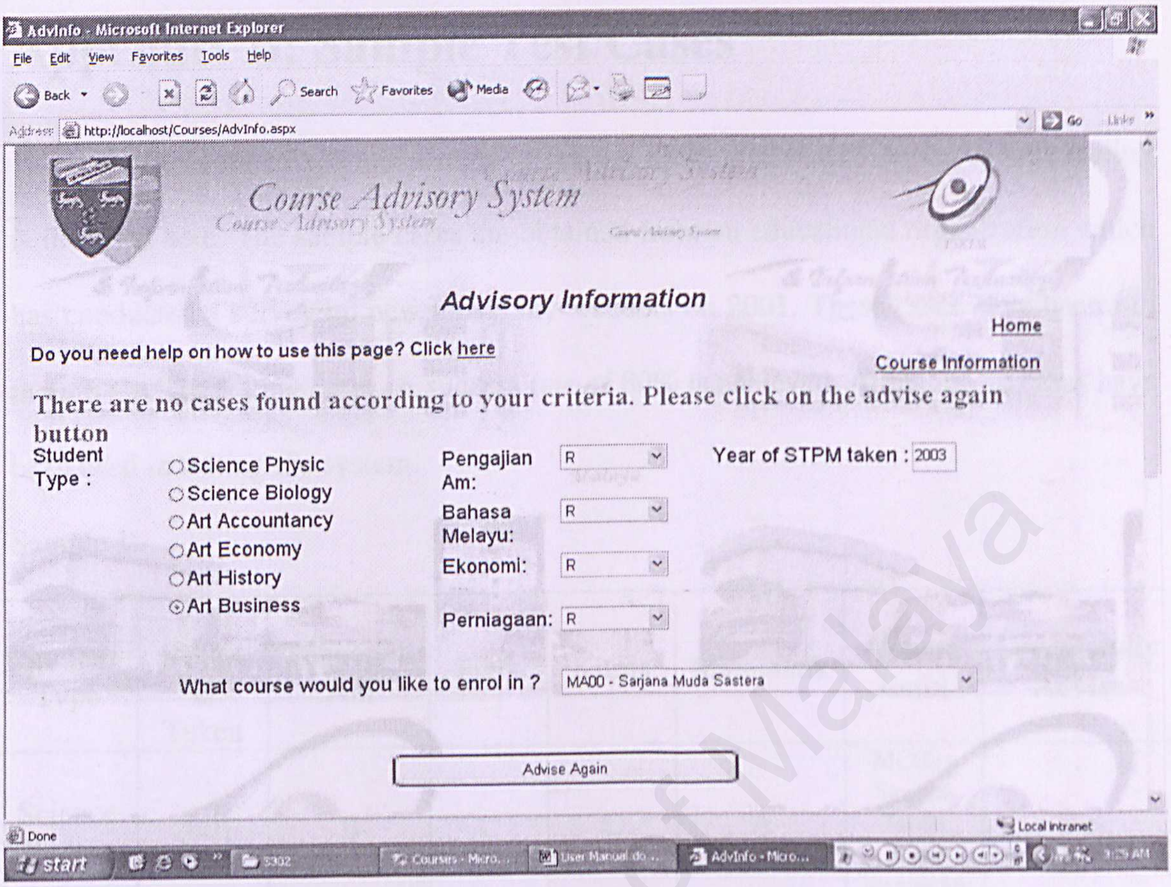
What course would you like to enrol in ? MA00 - Sarjana Muda Sastera

Course Advise	Similarity
MK11 - Sarjana Muda Kejuruteraan Telekomunikasi	99.99231921348746153846153846
MK04 - Sarjana Muda Kejuruteraan Mekanik	96.14616536733363076923076923
MK10 - Sarjana Muda Kejuruteraan CAD/CAM	92.300011521179800
MK02 - Sarjana Muda Kejuruteraan Elektrik	88.45769806828221538461538462
MC00 - Sarjana Muda Sains Komputer	80.76154998271826153846153846

Figure 4: Courses the students are being advice

A similarity higher than 95% is considered excellent as user may have a higher change of enrolling in.

If there is no match made, a message will be displayed as shows in figure 5. User can use the "Advise Again" button to obtain a lenient result although these courses will have a lower similarity. This means that the chances of user can enrol in these courses is low and it should serve as reference only.


AdvInfo - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back Search Favorites Media
Address http://localhost/Courses/AdvInfo.aspx
Go Links
Course Advisory System
Done
Local intranet
start
Courses - Micro...
User Manual.do...
AdvInfo - Micro...

Advisory Information

Home

Course Information

Do you need help on how to use this page? Click here

There are no cases found according to your criteria. Please click on the advise again button

Student Type :
○ Science Physic
○ Science Biology
○ Art Accountancy
○ Art Economy
○ Art History
● Art Business

Pengajian Am: R

Bahasa Melayu: R

Ekonomi: R

Perniagaan: R

Year of STPM taken : 2003

What course would you like to enrol in ? MA00 - Sarjana Muda Sastera

Advise Again

Figure 5: Error message when there is no match found

Appendix B: Sample Test Cases

In this appendix, some sample of the test cases that is used during system testing is included here. The sample cases are obtained from an educational organization which has conducted a survey on new university students on 2001. These cases have been run though CAS as testing cases. A success rate of 80% is achieved. A total of 30 cases have been used in testing the system.

Sample 1:

Student Type	Year STPM is Taken	Pengajian Am	Fizik	Kimia	Matematik	Course Enrol	Successfully Advise
Science Physic	2001	B	C	C	A	MC00 - Sarjana Muda Sains Komputer	Yes

Sample 2:

Student Type	Year STPM is Taken	Pengajian Am	Fizik	Kimia	Matematik	Course Enrol	Successful ly Advise
Science Physic	2001	A	B	B	A	MC00 - Sarjana Muda Sains Komputer	Yes

Sample 3:

Student Type	Year STPM is Taken	Pengajian Am	Biologi	Kimia	Matematik	Course Enrol	Successfully Advise
Science Biology	2001	A	A	A	A	MF00 - Sarjana Muda Farmasi	Yes

Sample 4:

Student Type	Year STPM is Taken	Pengajian Am	Biologi	Kimia	Matematik	Course Enrol	Successfully Advise
Science Physic	2001	A	C	C	C	MS00 - Sarjana Muda Sains	Yes

Sample 5:

Student Type	Year STPM is Taken	Pengajian Am	Biologi	Kimia	Matematik	Course Enrol	Successfully Advise
Science Biology	2001	A	C	C	D	MS00 - Sarjana Muda Sains	Yes

Sample 6:

Student Type	Year STPM is Taken	Pengajian Am	Bahasa Melayu	Sejarah	Ekonomi	Course Enrol	Successfully Advise
Art History	2001	A	C	C	D	ML00 - Sarjana Muda Undang - Undang	Yes

Sample 7:

Student Type	Year STPM is Taken	Pengajian Am	Bahasa Melayu	Sejarah	Ekonomi	Course Enrol	Successfully Advise
Art	2001	A	C	C	D	MA00 -	Yes

History						Sarjana Muda Sastera	

Sample 8:

Student Type	Year STPM is Taken	Pengajian Am	Perniagaan	Ekonomi	Matematik	Course Enrol	Successfully Advise
Art Economy	2001	B	C	D	C	MA00 - Sarjana Muda Sastera	No

Sample 9:

Student Type	Year STPM is Taken	Pengajian Am	Perakaunan	Ekonomi	Matematik	Course Enrol	Successfully Advise
Art Accountancy	2001	B	B	A	C	MA00 - Sarjana Muda Sastera	No

REFERENCES

Goh Shu Kea. (2002). *Case-Based Legal Expert System*. B.Comp.Sc Thesis. University of Malaya.

Hall, Marty. (2000) *Core servlets and java server pages.* New Jersey: Prentice Hall.

Karl A., Danny A., Timothy B. & Carl B. (2000) *Professional JSP* (1st Ed.). UK: Wrox Press Ltd.

Kolodner, J.(1993). *Case-Based Reasoning*. Morgan Kaufmann Publishers.

Payne, C. (2003). *Teach Yourself ASP .NET*. 2nd ed. Sams Publishing.

Sellappan, P. (1999) *Access 2000* (1st Ed.). Selangor: Federal Publication.

Sommerville. I. (2001) *Software engineering* (6th Ed.) Addison Wesley.

Whitten, J. L., Bentley, L. D., Dittman, K. C. (2002) *Systems Analysis and Design Methods*. 5th ed. McGraw Hill.

[1] http://demo.ctg.albany.edu/publications/reports/survey_of_sysdev?chapter=5

[2] http://brain.web-us.com/brain/ai.html

[3] http://artificialintelligence.ai-depot.com/introduction.htm

[4] http://www.15seconds.com/focus/A.I..htm

[5] http://www.mymajors.com

[6] http://www.cbr-web.org/CBR-Web/cbrintro/?info=index&menu=ai

[7] http://www.qub.ac.uk/mgt/intsys/advis.html

[8] http://www.java.sun.com

[9] http://www.math.ohiou.edu/~vnguyen/papers/aaa.pdf

[10] http://www.cs.tcd.ie/cbrpws/Papers/venturinir.pdf

[11] http://fit-bscw.gmd.de/pub/bscw.cgi/d33335856-2/*/*/MTSS2002-Ricci-final.pdf

[12] http://csdl.ics.hawaii.edu/~johnson/413s02/ProcessModels.pdf