CHAN CHONG WAH
WEK 990222

WXES3182
INTELLIGENT SCORE AGENT
WEB MONITORING & RETRIEVING SYSTEM

ABSTRACT

This project documentation is an exercise submitted to the Faculty Computer Science and Information Technology, University of Malaya, as a report for the Final Year Project Level One, WXES3181 and Final Year Project Level Two, WXES3182, which in turn serves as a partial fulfillment for the Bachelor of Computer Science degree.

The system proposed is ScoreAgent, an intelligent agent which will monitor a particular soccer Web site. The site is providing the online result to surfers for the on going soccer matches. If the result section in the page has changed, ScoreAgent will report the result of a soccer match live by sending a short message to the user's hand phone. This system is specially dedicated to all the soccer fans like me where they can't catch the live match every time their favorite team is playing.

In the beginning, the agent will be introduced and its objectives will be defined. The following section will identify some state-of-art AI technologies, programming language and additional AI tools that are considered in implementing the agent to have the intelligence features. Then, the methodologies will be discussed. Next, the result of analysis and design will be presented in the final chapter of the proposal. Then, the system development, implementation and testing phases will be discussed. Finally, the evaluation and conclusion on the agent. The chapter will summarize the achievements of the agent towards the goal that we have set early in this project.

WXES3182 is the next advancement stage of the Final Year Project Level One, WXES3181 the actual coding and development in WXES3182 are based on this proposal in WXES3181. However, there have been some changes to the system design since the WXES3181 proposal. The changes have been reflected in the product and in this report.

ACKNOWLEDGEMENT

First and foremost, I would like to express my utmost gratitude to my project supervisor, Miss Norisma Idris for her guidance, advice and encouragement. More importantly, she has given me a great opportunity to develop a system which interested me and the full freedom towards developing my proposed project. Thank you for letting me do what I want to do in WXES3181 and WXES3182. I had fun during the development of this system.

I would like to thank my project moderator, Assoc. Prof. Dr. Selvanathan for being so informative during the VIVA session. Best wishes for his recovery from illness. Also, my second moderator, Pn. Rohana Mahmud who willing to replace Dr. Selva and performing a good job on behalf of Dr. Selva. The advice and feedback were so useful for my project.

I would also like to extend my thanks and appreciation to all my friends and course mates who have been so helpful and cooperating in providing me some crucial thoughts and advice.

Last but not least, I would like to give my thanks and loves to my family.

I would like to dedicate the success of this system to Liverpool Football Club. With my love towards this club, I aimed to build the ScoreAgent.

DISCLAIMER

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

CHAPTER 1
INTRODUCTION

This introductory chapter gives a description or purpose of the project and problems to be solved. The significance and rationale of the project will be discussed here. Furthermore, the system functions, limitations and its assumptions will also be dwelt into later in this chapter.

1.1 PROJECT DEFINITION

Human beings have tried to avoid tedious works. Everybody does. Therefore, we often seek services from an agent. In real life, we delegate an agent to perform tasks for us, tasks that we are not specialized in or tasks that we are not willing to do by ourselves. The same situation is happening in the world of computer after the concept of "agent" emerged and has been greatly discussed only recent years.

The domain of this project is on the sports area, more specifically, soccer. The agent called ScoreAgent will monitor a Web site and sends the latest score in a text format via SMS (Short Message Service) to the user's mobile phone. During the process, the system will need to cooperate with mail-server and Web-server towards achieving goal. The system is trying to apply the most cutting edge Web technologies and the convenience of Short Message Service (SMS) into the life of the soccer fans.

1.2 OBJECTIVE

There are some reasons or objectives which this system has been proposed.

a) An alternative way to get the latest result

Most of the Malaysians are watching English Premier League, a soccer league in England. There are numerous fans for each team in the league. As all the soccer fans have experienced, there is seldom a live telecast match on TV these days from abroad in Malaysia. Some soccer maniacs are anxious about the result especially when their team is having a vital match with their opponent. They will either subscribe to a private broadcasting company to watch the live match or they go to the Web sites and check for results. The agent will provide them the third alternative way to get the latest result on their hands and it is free.

b) Applying AI techniques in agent & Utilizing SMS in the world of soccer

The main objective of this project is to develop an agent that is intelligent enough to monitor some particular sport Web pages on their result section. If there is any changes on the HTML contents, it will access and parsing the HTML codes to search for the latest result, formats the text and sends a message (SMS) to notify user about the latest result. The character of autonomous of the agent will make it intelligent enough to organize and administrate itself to perform the tasks on behalf of the user. Some equivalent inference techniques will be applied. The user just has to pass some initial parameters to the agent and the agent will do the rest works by itself. The user will only interact with the agent if the user wants to change the parameters (via internet). These parameters might determine when the agent should perform their monitoring tasks or the user can asks the agent to work on predefined period. This will ease the working hour of the agent.

c) To make agent a common

It has been an immense number of agents in the world of software either in a standalone system or Internet. Therefore, it is a need to introduce the agent concept to others because it has been seen as the part of the future.

1.3 SCOPE

The main function of this system is to provide an agent to send a message whenever the score for a particular match is updated on the Web page. This might need some configuration on a mail server where it will allow the agent to send a short mail to the telco company and the company will send SMS to their subscribers. The agent should be resided in a server where it can monitor the Web site frequently.

1.4 SYSTEM LIMITATION

The idea of this project is to send the latest soccer match result to user's mobile phone via SMS server which is provided by some telecommunication companies. Locally, there are only 2 companies that allowing their users to receive messages or short email via the Internet at the moment. The concept of sending messages is similar to sending an email to another. The example of sending is 0126599904@sms.maxis.net.my for Maxis users and 0193232563@sms.celcom.com.my for Celcom users. Both companies are providing GSM service as their telecommunication platform.

Therefore, the arriving time of the messages (result) will be very much depending on the SMS server's (telco side) performance. If the agent takes actions and sends the result during the peak hour of the server, this will result the delay of the message to the user. But the good news is, often the soccer matches in England are at late night so the time delayed is not severe.

1.5 PROJECT SCHEDULE

	1	2	3	4	5	6	7	8	9	10++
PHASE I										
Getting briefing from lecturer and propose the topic.	▬									
Identifying objective of the proposed topic.	▬	▬								
Estimate scope and the specific domain where the intelligent agent should applied in.	▬	▬								
Searching information from various sources about intelligent agents from the Internet and books.	▬	▬	▬	▬	▬	▬	▬	▬		
Analyzing and research the information.					▬	▬				
Determine system module.						▬	▬			
Design the recommended system.							▬	▬		
Presentation of the proposal.									▬	
Documentation	▬	▬	▬	▬	▬	▬	▬	▬	▬	▬
PHASE II										
Develop system.										
Documentation.										▬
Testing and maintaining the designed system.										▬
Implementation of the system.										▬
Presentation.										▬

Prepared by,

Chan Chong Wah

CHAPTER 2

LITERATURE REVIEW

Research and literature reviews were the tasks to be done in this chapter. Effort to dig deeper for information about the proposed system is important. With that, the overview of the system will be seen. The materials below are concluded following some efforts such as studying and looking for the similar projects from FSKTM document room, looking for reference books from library, extra reference materials from Internet and purchasing some useful related books.

This chapter will cover the topics on intelligent agent and artificial intelligence approaches, Web development and applications technologies, appropriate programming languages and DBMS. The analysis of the current available system also covered briefly.

2.1 ARTIFICIAL INTELLIGENCE

Artificial intelligence is a subdivision of computer science devoted to creating computer software and hardware that imitates the human mind. The main goal of AI is to make computers smarter by creating software that will allow a computer to mimic some of the functions of the human brain in selected applications. The idea is not to replace the human beings, but to provide us a more powerful and hopefully safer tools in assisting our works [1].

Computers already emulate some of the simpler activities of the human mind. They can perform mathematic calculations, manipulate numbers and letters and make decision on behalf of the human. Now, they have become our agent!

2.1.1 *Intelligent Agent*

The explosive growth in the Internet and distributed computing has led to the idea of applying an agent within the Internet domain. On the Internet, an intelligent agent (sometimes called bot) is an autonomous program that perceiving its environment and acts upon that environment [2]. Then it gathers useful information that interested us without our presence or interference. The result will be presented to us daily or according to our desired period. Therefore, perceiving the environment is important to an agent.

2.1.1.1 Characteristic of an Agent

1) Perception

It is often a term to describe an intelligent object. As for animals, they perceive their environment by senses. They sense through smell, taste, hearing and sight. In term of intelligent agents, they sense their environment through their sensory. They will filter out and ignore inputs which are expected and usual. They learn to focus when there is an unexpected change in their environment. A very good real life example is when we are in the car, we can still drive safely (most of the time) while we are using hand phone but when there is a car suddenly appear close to our car, we have switch our focus immediately to the situation and react against it. The reaction for this situation is, we keep our car far away from the car. If we have not perceiving the environment well, an accident will happen. The information comes in through agents' sensory. It can be a message about the environment from the system or other agents. If an agent resided in an email or newsgroup monitor domain, it has to know how to recognize a new arrived mail whether to inform the user or not. It should only inform the user if it knows that the mail is important or urgent to the user. The more autonomous agent can pursue agenda independently of its user; take preemptive or independent actions that will eventually benefit the user. This should be a behavior for human beings too. As an example, a user wants a ticket for football match on Sunday; the agent will arrange it for user [3].

How does an agent know when to perform an action? This is where we have to deal with the events, conditions and actions [3].

2) *Events*

An event is anything that happens to change the environment which it should perceiving and monitoring from time to time. An agent should always alert. It could be a time when the agent should remind us about an occasion, content of a particular Web page has changed, our favorite books are available on the Web or even display the result of auction. These sample events will trigger the next stage before the agent makes the decision to notify us, evaluating and recognizing condition.

3) *Conditions*

When an event occurs, the agent has to evaluate and respond to it. After determining based on some conditions, the agent will take consideration on the event whether it will bring us to the next stage (action). As an example in mail notification agent, it should be able to recognize the mail's category and label it, either the mail is urgent or a bulk mail. If the mail is labeled URGENT, then the agent should notify us instantly (an action) else it could ignore the event and stay idle.

4) *Actions*

In real living life, we go for agents because we want them to do something on behalf of us. An important notion in intelligent agent is delegation. We can not delegate a task to someone who we think they could not have the task done. Therefore, we must put faith and trust on them although they might mess up things. This is just the same situation on why we applying intelligent agents in the Internet. We want them to complete tasks for us but we have to weigh the risks and rewards. Therefore, we should assume that the agent would act and think rationally (intelligent) before it takes actions. It will treat our interest seriously.

When a significant event (unusual and unexpected event which matches the condition to take action) occurred, an action will be performed through the agent's *effector* (muscle). As an example, the agent will send a notification to user about the arriving of an urgent mail in his mailbox. An agent works at their best when they could exhibit graceful degradation in cases of a communications mismatch (the 2 parties did not communicate well, causing ambiguity) or a domain mismatch (inappropriate result). If most of a task can still be accomplishes, instead of failing to accomplish any of the task, this is generally a better outcome, and gives the user more trust in the agent's performance.

2.1.1.2 Taxonomies of Agents

There have been many ways suggested to classify and categorize agents. The more general way is to place the agent in the context of intelligence, agency and mobility[3].

1) Agency

The agency deals with the degree of autonomy. An agent represents the user, helps the user, guides the user and takes actions on behalf of the user. As an example, the agent operates itself on the Internet while user is disconnected from the Web. During this period, agent will continue to monitor the Web content. If there is a change and the conditions match the goal, then an appropriate action will be taken.

2) Intelligence

Intelligence refers to the ability of the agent to apply the domain-specific knowledge and processing to solve problems. The agent can apply some sophisticated AI-based method to inference and learn upon the domain which they should alert. A learning agent will able to adapt to its environment, keeping in charge and control of the environment.

3) Mobile

An agent is considered mobile when it able to move between systems in a network or outside the network. Sound like no restriction? This actually poses some issues of security and privacy in the networks. Therefore, intranets are one particular ripe environment where the agent can perform their unique behavior, mobility, due to less security compared to the wide-open Internet. A more mobile agent may be composed on one computer and then sent to another to gather date. It then returns the data to the client and is sent out to gather another piece of information. This pattern repeats until the agent is exhausted the search parameters or the user stops it.

2.1.1.3 Differences between Agents and Distributed Objects

"What is the big deal with agents? They are just distributed objects with a different name. COBRA (Common Object Request Broker Architecture) has been around for years, Microsoft's (OLE, ActiveX, COM, DCOM, etc.) objects have been around for years too. What is the difference between agents and object?" [3]

That is most probably the argument that skeptics pointed out. A distributed object application is defined by the objects, the data and the behavior (methods) required to implement the function needed. The interactions between objects are explicitly defined; the sequence of method calls is spelled out in gory detail. An object is a software entity that encapsulates state (data) and behavior (function) and exposes a set of methods or procedures to manipulate the state. Objects are used by other objects to perform actions. Objects don't initiate actions of their own volition.

Whereas a multiagent system has a collection of software entities that autonomously perform actions. Each agent has a complex internal state than an object. Furthermore, each of the agents has their own internal goals. Objects are some coded software to perform some specific roles in the program or system. They are the authority themselves. Agents can decide on what they want to do and when they want to perform. The decisions are on their hands, depending on the current situation or state before they make the decisions. They can say "no" when there is a request from other agents.

In a distributed object application, every object is contributing a small piece of function in achieving a single application goal. In an agent system, a collection of goal-oriented software entities cooperates to achieve a single application goal. Objects invoke methods. Agents have conversations among themselves. Objects are data packets with buttons waiting to be pushed. Agents are actively deciding what buttons to push.

2.1.1.4 Multiagent

Most of the cases, a task is completed faster and more effective when more than one people are working towards the problem. A question is raised. "Why we let an agent work on a goal while we can ask many agents work on the same goal?" This is where the term, multiagent comes in. The unique characters owned by a multiagent are [3]

- Each agent only has a limited view of the state of the world.
- No global control
- Decentralized data
- Computation (in agent) is asynchronous
- Speak the same language and using the same symbol (knowledge representation)
- Talk or write at different time
- Planning must occur between agents

2.1.1.5 Constructing an Intelligent Agent

How can we build an agent which will suit the user's needs and requirement? In what kind of domain or knowledge representation technique we should apply in construction an agent? [3]

a) When the agent has less number of situations to respond, we only need to hard-coding the intelligence algorithms into procedural codes.
b) If the agent has to solve problems at different level of abstraction, frame or semantic nets is the ideal solution.
c) If agent has to answer some questions and generate new facts from existing data, we should go for predicate logic or if-then rules.
d) If the agent will interact with other agents and share the knowledge among themselves, KIF leads the way.
e) When there is a large amount of uncertainty in problem domain, Bayesion network and if-then rules with certainty factor are the perfect match.
f) When the agent required to find an optimal answer towards a situation, state-based searching techniques or genetic algorithm might be the answer.

Therefore, a programmer needs to craft a solution that meets the needs of a specific application.

2.1.2 *Knowledge Representation*

An artificial intelligence program usually has knowledge about the domain of interest. The process of collecting and organizing the knowledge is called knowledge engineering. It is the most time consuming part in developing an intelligent program. At the moment, knowledge representation methods share some common characteristics. First, they can be programmed with existing programming languages. Second, they are designed so that the facts and other related knowledge within could be used in reasoning. This means that the knowledge base contains a data structure that can be manipulated by an inference engine. The inference engine will use some searching and pattern matching techniques that would answer questions and give a decision or conclusion to the subjected matter. The question is, how to represent knowledge in computers? The answer had found is symbols [1].

2.1.2.1 *Predicate Logic*

In the predicate calculus, a proposition or premise is divided into two parts, the arguments (or objects) and the predicate (or assertion). In a common English language sentence, objects and individuals are nouns that serve as subject and object of the sentence. In a sentence the predicate would be the verb or part of the verb. The two combined to create a proposition:

PREDICATE (object1, object2)

As an example, the sentence "Malaysia is hot in all season" can be represented can be represented in predicate logic several ways.

Single parameter:

place(Malaysia) and temperature(hot) and season(all)

or

Single relation:

hot(Malaysia, allseason)

Both statements could mean the same as facts in the knowledge base. Two techniques, called resolution and unification, are used to process predicate statements to prove whether a particular statement is true or not, based on other known facts. These has formed the basis for Prolog (Programming in Logic).

2.1.2.1.1 Resolution

Resolution is an algorithm for proving facts true or false by virtue of contradiction. If we want to prove a theorem X is true, we have to show that the negation of X is not true.

Observe the following statements,

1) not feathers(Tweety) or bird(Tweety)
2) feathers(Tweety)
3) not bird(Tweety)
4) bird(Tweety)

From the statements above, it is clear that sentences 3 and 4 cannot both be true. The condition will be either Tweety is a bird or else. We have a contradiction! We have just proven that our first assumption, not bird(Tweety), is false, and the alternative, bird(Tweety), must be true. If the clauses to be resolved are selected in systematic ways, then resolution is guaranteed to find a contradiction if one exists, although it may take a long time to find.

2.1.2.1.2 *Unification*

It is a technique for taking two sentences in predicate logic and finding a substitution that makes them look the same. This is a requirement for proving theorems using resolution, as discussed. if two predicates are identical, then they match, by definition.

Observe the following statements,

1) hates(X,Y)
2) hates(George, broccoli)
3) hates(Alex, spinach)
4) hates(X, vegetable(Y))
5) hates(George, vegetable(broccoli))
6) hates(Z, broccoli)

We could unify sentence 6 with sentence 1, 4 with 5.

2.1.2.2 Semantic Network

One of the methods in knowledge representation schemes is the semantic net which are some graphical depictions of knowledge that show hierarchal relationship between objects. A simple semantic network is made up of number of nodes (can be concepts, events or actions) which represent objects and descriptive information about objects. Objects can be any physical item around us. Arcs connect the nodes to each other. The arcs show the relationships between the various objects and descriptive factors. Is-a usually shows that an object belongs to a larger class. Has-a is used to identify characteristics or attributes of the object nodes. Semantic net shows the behavior of inheritance too. An example of semantic net shown below.

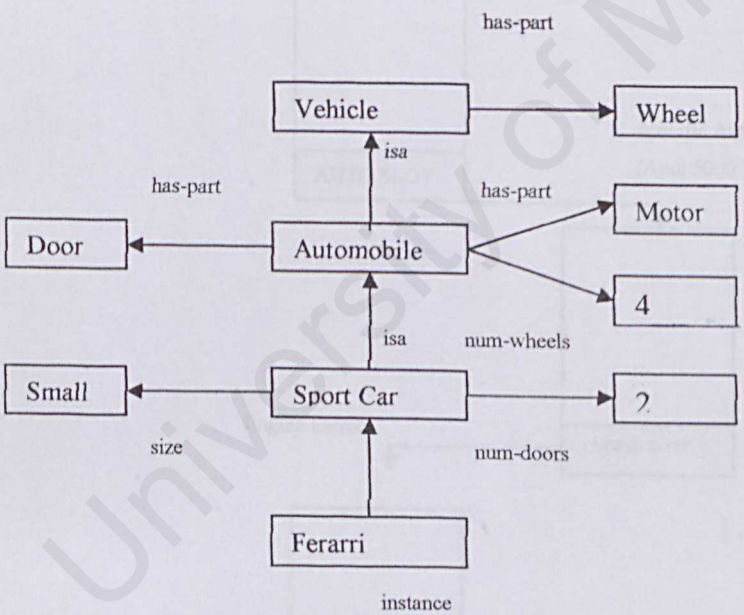


Figure 2-1: An example of semantic net

2.1.2.3 Frames

In AI, frames are called slot-and-filled data representations. The slots are the data values and the fillers are attached procedures. Frames are linked into a hierarchy to represent has-part and isa relationships. It is a collection of attributes that define the state of an object and its relationship to other frames (objects).

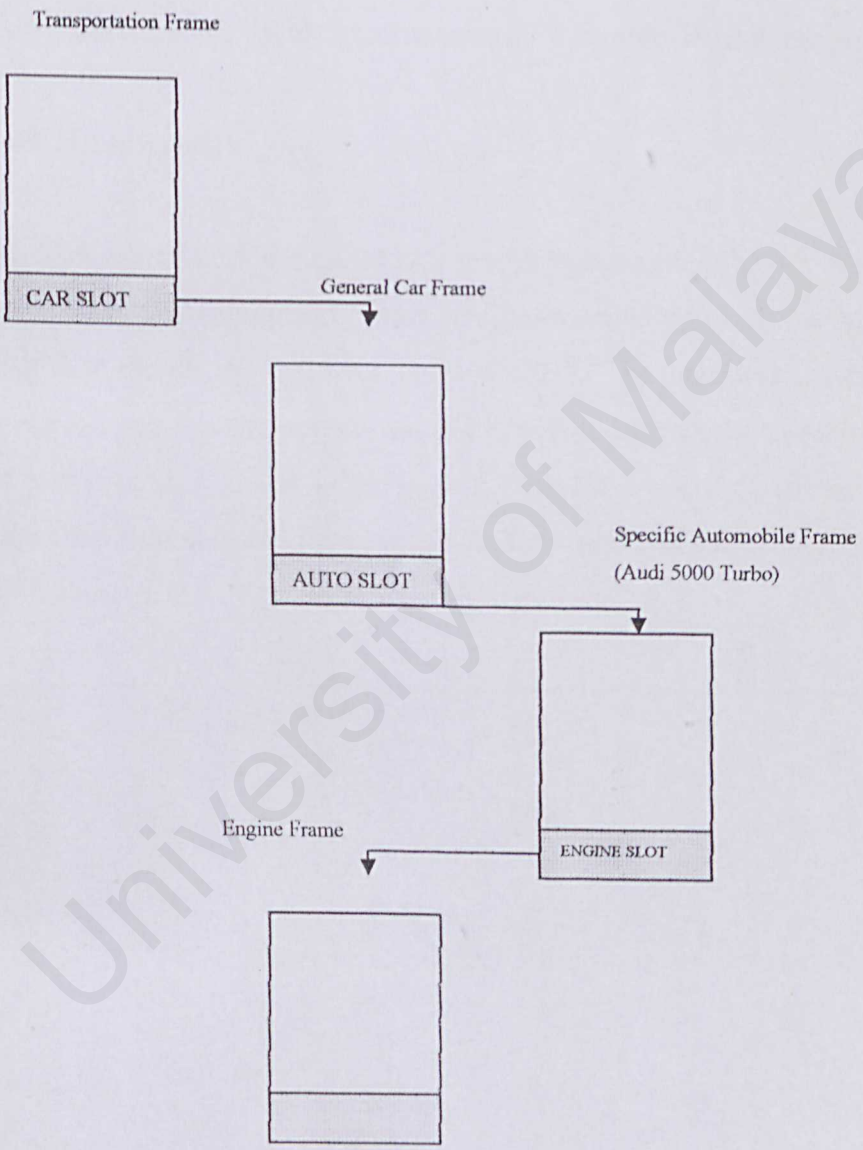


Figure 2-2: An example of frames.

Knowledge represented in a hierarchy of frames that inherit properties from higher-level frames.

The slots (data values) contain the default values. For example, a default value for number of wheels could be four. From the Figure 2-2, the gray colored fillers (attached procedures) permit new information to be added on.

2.1.3 *Problem Solving Using Search*

In the field of artificial intelligence, it is important to know what problem you are trying to solve. How do you can represent define the problem so that a computer understand and thus solve it? How do you define the problem with enough precision so that you can figure out how to represent it? Although the search-based methods are not the best technique to demonstrate how does our brain solves problems but the methods have proven extremely useful as one of the most successful technique in problem solving.

2.1.3.1 Breadth First Search

This technique will examine all the nodes in a search tree, beginning with the root node. The nodes in each level are completely examined before moving on to the next level. A simple breadth first search is illustrated in Figure 2-3. The numbers inside the node circles designate the sequence in which the nodes are examined. In thins instance, the search would end at node 7 as that is the goal. A breadth first search of the state space will always find the shortest path between the initial state and the goal state, with the least number of steps.

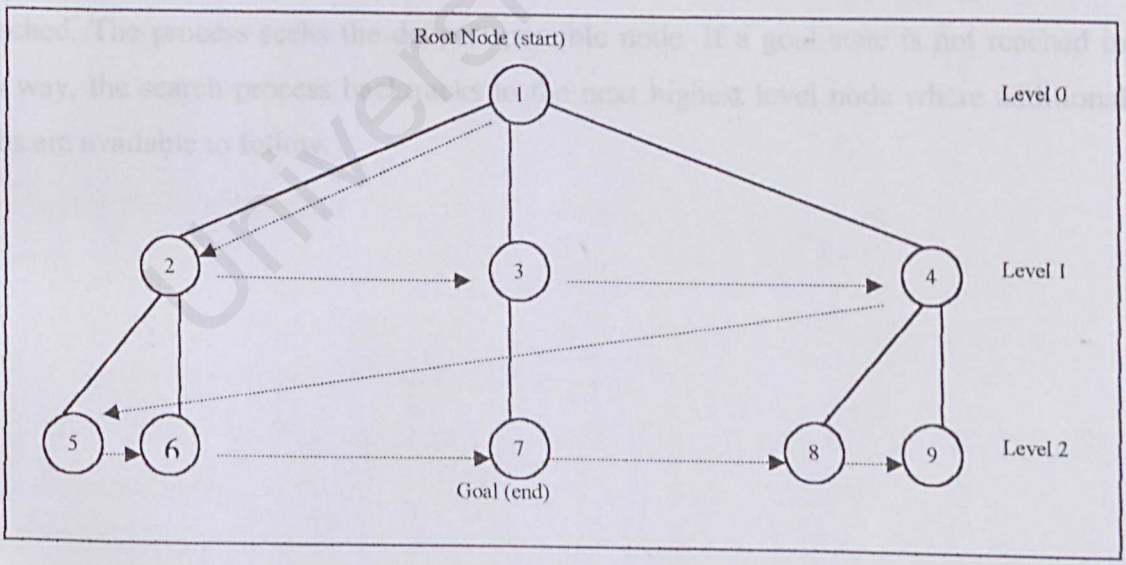


Figure 2-3: A breadth first search.

The algorithm defines a way to move through the tree structure, examining the values at nodes in a controlled and systematic way, so that we can find a node that offers a solution to the problem we have represented using the tree structure [3].

1. Create a queue and add the first *SearchNode* to it.
2. Loop:
   a) If the queue is empty, quit.
   b) Remove the first *SearchNode* from the queue.
   c) If the *SearchNode* contains the goals state, then exit with the *SearchNode* as the solution.
   d) For each child of the current *SearchNode*, add the new *SearchNode* to the back of the queue.

2.1.3.2 Depth First Search

It begins at the root node and works downward to successively deeper levels. An operator is applied to the node to generate the next deeper node in sequence. In other words, a parent node generates a child or successor node. This process continues until a solution is found or backtracking is forced by reaching a dead end. Figure 2-4 shows a simple depth first search. The numbers inside the nodes designate the sequence of nodes generated or serached. The process seeks the deepest possible node. If a goal state is not reached in this way, the search process backtracks to the next highest level node where additional paths are available to follow.

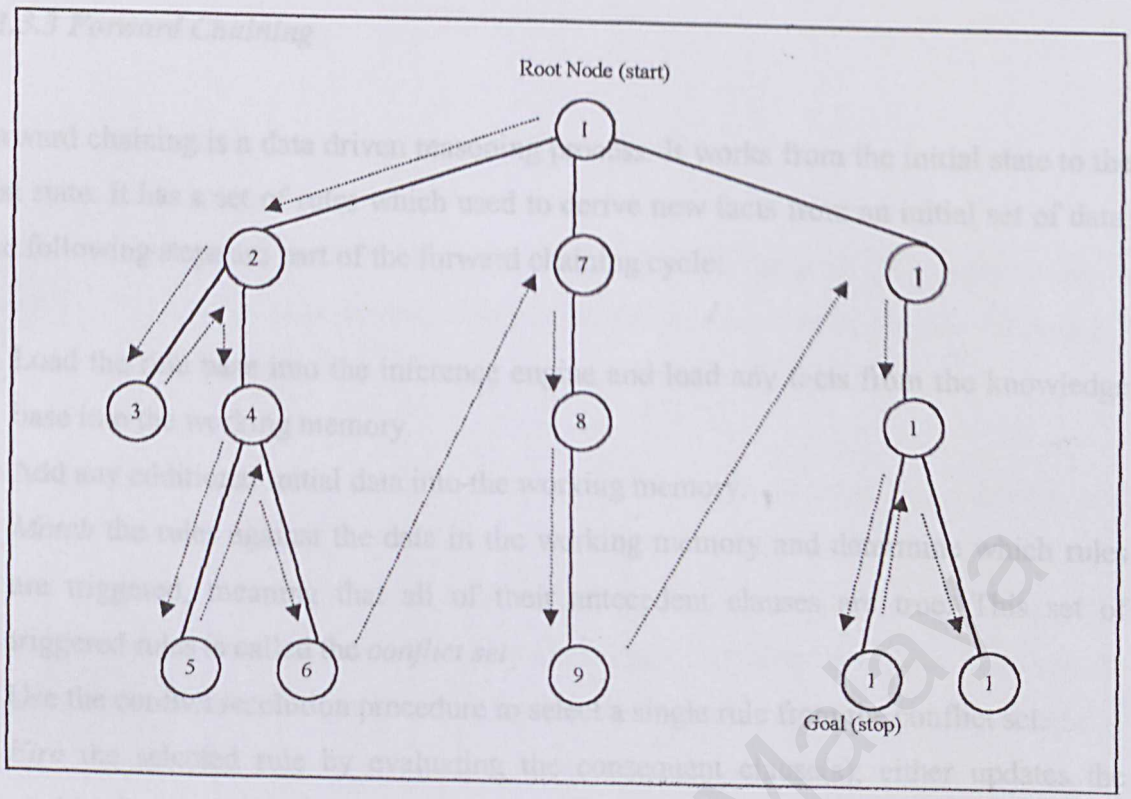


Figure 2-4: A depth first search

Depth first search is another way to systematically traverse a tree structure to find a goal or solution node. Instead of completely seracing wach level of the tree before going deeper, the depth first algoritm follws a single branch of the tree doen as many levels as possible until it either reaches a solution or a dead end [3].

1. Create a queue and add the first *SearchNode* to it.
2. Loop:
   a) If the queue is empty, quit.
   b) Remove the first *SearchNode* from the queue.
   c) If the *SearchNode* contains the goals state, then exit with the *SearchNode* as the solution.
   d) For each child of the current *SearchNode*, add the new *SearchNode* to the front of the queue.

2.1.3.3 Forward Chaining

Forward chaining is a data driven reasoning process. It works from the initial state to the goal state. It has a set of rules which used to derive new facts from an initial set of data. The following steps are part of the forward chaining cycle:

1) Load the rule base into the inference engine and load any facts from the knowledge base into the working memory.
2) Add any additional initial data into the working memory.
3) *Match* the rules against the data in the working memory and determine which rules are triggered, meaning that all of their antecedent clauses are true. This set of triggered rules is called the *conflict set*.
4) Use the conflict resolution procedure to select a single rule from the conflict set.
5) *Fire* the selected rule by evaluating the consequent clause(s); either updates the working memory if it is a fact-generating rule, or calls the effector procedure, if it is an action rule. This is referred to as the *act* step.
6) Repeat steps 3, 4, and 5 until the conflict set is empty.

2.1.3.4 Backward Chaining

Backward chaining is called goal driven. The process starts at the goal state and using special inverse operators to generate successor nodes to reach the initial state. How we can identify the goal node as the starting point? If we know the goal, we wouldn't have to search for it. Actually, in backward chaining we simply pick one of the several possible goal nodes and attempt to prove or justify it by searching backward.

Backward chaining is used for advisory expert systems, where users ask questions and get asked leading questions to find the answer. One advantage of backward chaining is because the inferencing is directed, information can be requested from user when it is needed. Some reasoning systems also provide a trace capability that allows the user to ask the inference engine why it is asking for some piece of information, or why it came to some conclusion. The following steps are part of the backward chaining cycle:

1) Load the rule base into the inference engine and load any facts from the knowledge base into the working memory.
2) Add any additional initial data into the working memory.
3) Specify a goal variable for the inference engine to find.
4) Find the set of rules that refer to the goal variable in a consequent clause. That is, find all the rules which set the value of the goal variable when they are fire. Put each rule on the goal stack.
5) If the goal stack is empty, halt.
6) Take the top rule off the goal stack.
7) Try to prove the rule is true by testing all antecedent clauses to see if they are true. We test each antecedent clause, in turn, as follows:
   a) If the clause is true, go on the next antecedent clause.
   b) If the clause is false, then pop the rule off the goal stack; go to step 5.
   c) If the truth value is unknown because the antecedent variable is unknown, go to step 4, with the antecedent variable as the new goal variable.

If all antecedent clauses are true, fire the rule, setting the consequent variable to the consequent value, pop the rule off the goal stack, and go to 5.

2.1.4 *Learning System and Neural Network*

One of the most significant features for an intelligence behavior is the ability of the system to learn from experience and be adaptive to the environment around it. The result of this unique behavior, the system will perform better on the given tasks and thus improvement of the overall performance and credibility. If the system can learn or be taught, it know when and how to avoid errors. It could choose the best "partner" to cooperate in completing a task. It remembers the mistakes and tries to not to do it again. It recognizes the environment. Experience is the key in learning system. By the time goes, the more mistakes it made, the more knowledge and experience it gained, and thus the more intelligent the system is. There are three major types of learning paradigms, which are supervised, unsupervised and reinforcement [3].

1) Supervised Learning

One of the common forms of learning is supervised learning. It also called as "programming by example". In this case, the agent is trained by given some examples of inputs, together with the desired outputs or actions. The agent will try to predict the outputs based on the inputs. If the result (output) is not satisfied, the agent is adjusted until it produces the desired output.

This process will be repeated over and over again until the accuracy of the result can be trusted and relied on. Historical data from databases, sensor logs, or trace logs is often used as the training or example data.

2) Unsupervised Learning

Unsupervised learning, sometimes called self-organization requires only the inputs to train the agent. During the training process, the agent has to recognize the similarities between inputs and identify the features from the input data. This is accomplished by the training algorithm that extracts statistical regularities from the training set, representing them as the values of weights.

3) Reinforcement Learning

Reinforcing learning is unlike the other two learning methods mentioned above, where agent is trained with sample outputs. Instead reinforcement learning uses punishment and reward to train the agent. This method is similar to human training an animal to perform some specific task by giving it reward or punishment. First, the agent is presented with a sample input and the agent computes what it thinks should be the sample output. Then the agent is supplied with a real valued judgment from the teacher. If the value is positive, then the agent will receives a reward and vice versa (reminds me of the sea lion when it gets a fish from its trainer in the circus). After this, the weights are adjusted and the process continues.

4) On-line and Off-line Learning

Under on-line learning, agent is sent out to perform some tasks. It may have to learn from each transaction which is in the process. This learning method can be seen as an on-the-job training and places severe requirements on the learning algorithms. It must be very fast and stable.

Mean while, off-line learning is like a business seminar. You send your agent into a training field before send it out to perform the tasks. After a suitable training on the agent, it can apply the skills and knowledge which it learnt during the "seminar" when it has placed in a new environment to continue its tasks.

2.1.4.1 Neural Network

Introduction

The fundamental processing element of a neural network is a neuron. This building block of human awareness encompasses a few general capabilities. Much is still unknown about how the brain trains itself to process information, so theories abound.

In the human brain, a typical neuron collects signals from others through a host of fine structures called dendrites. The neuron sends out spikes of electrical activity through a

long, thin stand known as an axon, which splits into thousands of branches. At the end of each branch, a structure called a synapse converts the activity from the axon into electrical effects that inhibit or excite activity from the axon into electrical effects that inhibit or excite activity in the connected neurons. When a neuron receives excitatory input that is sufficiently large compared with its inhibitory input, it sends a spike of electrical activity down its axon. Learning occurs by changing the effectiveness of the synapses so that the influence of one neuron on another changes. Figure 2.5 below show a biological neuron in a human brain.

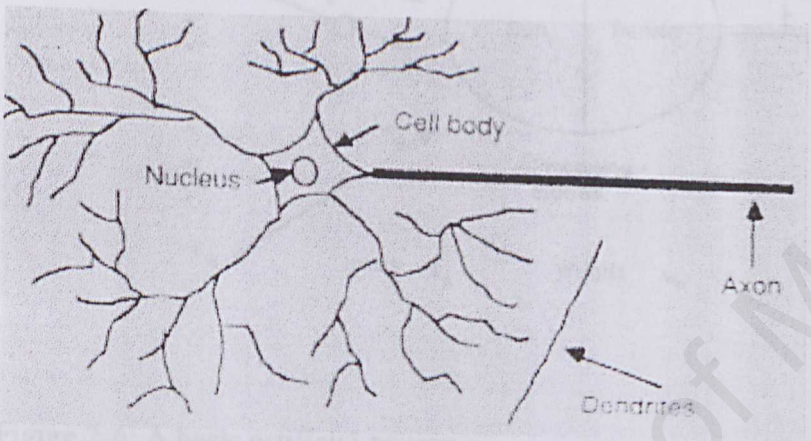


Figure 2-5: A biological neuron

In information technology, a neural network is a system of programs and data structures that approximates the operation of the human brain. A neural network usually involves a large number of processors operating in parallel, each with its own small sphere of knowledge and access to data in its local memory. Typically, a neural network is initially "trained" or fed large amounts of data and rules about data relationships (for example, "A grandfather is older than a person's father"). A program can then tell the network how to behave in response to an external stimulus (for example, to input from a computer user who is interacting with the network) or can initiate activity on its own (within the limits of its access to the external world). In making determinations, neural networks use several principles, including gradient-based training, fuzzy logic, genetic algorithms, and Bayesian methods.

This artificial neural network is comprised of various neurons connected by its weight. These neurons are modeled after the biological neuron shown above. Figure 2-6 show the architecture of a basic artificial neuron.

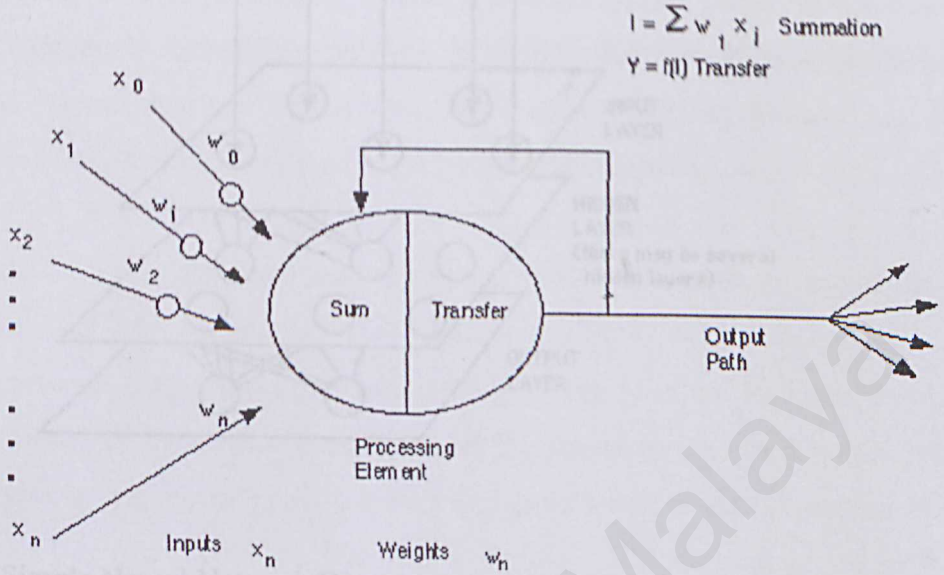


Figure 2-6: A basic artificial neuron.

An artificial neuron is a basic processing element of a neural network. Refer to the above figure the various inputs to the network is represented by X (n). This input is analog to the dendrites of a biological neuron. Each of this input is multiplied by a connection weight W (n). This product will be summed together and send to a transfer function to generate a result. There are many different transfer functions, which can be used such as hard limit, pure linear, log-sigmoid, etc.

The neural network can consist of a lot of layers. A single layer is combined of one or more artificial neurons. Figure 2-7 below shown a simple neural network with three layers namely input layer, hidden layer and output layer [4].

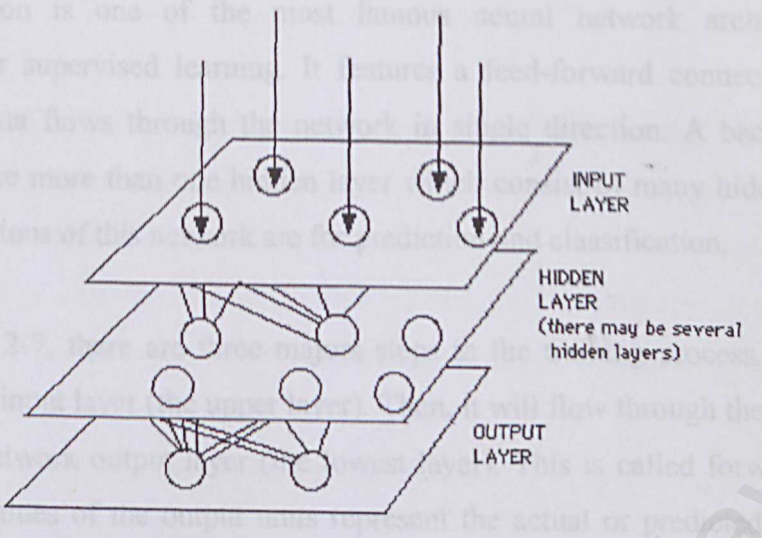


Figure 2-7: A Simple Neural Network Diagram

2.1.4.2 Back Propagation

Back Propagation is one of the most famous neural network architecture being implemented for supervised learning. It features a feed-forward connection topology, meaning that data flows through the network in single direction. A back propagation network can have more than one hidden layer which consist of many hidden units. The primary applications of this network are for prediction and classification.

Refer to Figure 2-7, there are three majors steps in the training process. Input data is presented to the input layer (the upper layer). Then, it will flow through the network until it reaches the network output layer (the lowest layer). This is called forward pass. The *activations* or values of the output units represent the actual or predicted output to the network. Together at the output layer, the desired output value is also presented to the network because this is a supervised learning, a sample of the desired output must be presented with the set of input data. The difference between the desired output and actual output network is computed, thus producing the error. The error will be passed backwards through the network to adjust the connection weights.

Each network input unit takes a single numeric value, x_i, which is usually scaled or normalized to a value between 0.0 and 1.0. This value becomes the input unit activation. When we propagate the data forward we compute the sum of the products of the input unit activations and the weights connecting those input layer for each unit in the hidden layer. The sum is passed through a nonlinear activation function, f, producing the unit activation, y_i, as shown in Figure 2-6. Therefore, the formula for computing the activation of any unit in a hidden or output layer in the network is

$$y_i = f(sum_j = \Sigma x_i w_{ij} + \theta_j)$$

Where I ranges over all units leading into unit j, and the activation function is

$$f(sum_j) = \frac{1}{1 + e^{-sum_j}}$$

As mentioned earlier, we use the S-shaped sigmoid or logistic function for **f**. For a range of sums from –5 to +5 we get a *y* value ranging from 0 to 1. Extremely large positive or negative sums get squashed (that's a technical term) into that same range. The effect of the threshold θ is to shift the S-shaped curve left or right. For example, if the threshold is +5, then the sum would have to be –5 in order to produce an output value of 0.5.

The formula for calculating the changes to the weights is

$$\Delta w_{ij} = \eta \delta_j y_i$$

Where w_{ij} is the weight connecting unit i to unit j, η is the learn rate parameter, δj is the error signal for that unit, and y_i is the output or activation value of unit i. For units in the output layer, the error signal is the difference between the target output t_j and the actual output y_j multiplied by the derivative of the logistic activation function.

$$\delta_j = (t_j - y_j) f'_j (sum_j) = (t_j - y_j) y_j (1 - y_j)$$

For each unit in the hidden layer, the error signal is the derivative of the activation function multiplied by the sum of the products of the outgoing connection weights and their corresponding error signals. So for hidden unit j

$$\delta_j = f'_j (sum_j) \Sigma \delta_k w_{jk}$$

Where k ranges over the indices of the units receiving units j's output signal.

A common modification of the weight update rule is the use of a momentum term α, to cut down on oscillation of the weights. So, the weight change becomes a combination of the current weight change, computed as before, plus some fraction (α ranges from 0 to 1) of the previous weight change. This complicates implementation because we now have to store the weight changes from the prior step.

$$\Delta w_{ij} = (n + 1) = \eta \delta_j y_i + \alpha \Delta w_{ij} (n)$$

The mathematical basis for backward propagation is described in detail in Rumelhart, Hinton, and Williams (1986). When the weight changes are summed up (or batched) over an entire presentation of the training set, the error minimization function performed is called gradient descent. In practice, most people immediately update the network weights after each input vector is presented. While pattern updates, as this is called, can sometimes produce undesirable behavior, it usually results in faster training than using the batch updates.

2.2 WEB TECHNOLOGIES AND JAVA

2.2.1 Web Page Languages

2.2.1.1 HTML

HTML – the Hypertext Markup Language is a markup language. It is used to format text and information, i.e. to identify elements of a Web page so that a browser, such as Microsoft Internet Explorer or Netscape's Communicator, can render the page on a computer screen [3].

A HTML file (HTML document) can be created using simple text editor such as Notepad. A HTML file usually ends with either the .htm or the .html file name extension.

A free HTML source code editor called HTML-Kit can also be downloaded at http://www.chami.com/html-kit. There are also software packages, such as Microsoft Frontpage, that can be used to create Web pages visually, without the need for the page developer to code with HTML directly. Microsoft Visual Interdev is a useful program that can be used for both simple HTML editing and more complex Web application development [5].

2.2.1.2 DHTML

Dynamic HTML is a collective term for a combination of new Hypertext Markup Language (HTML) tags and options, that will let you create Web pages more animated and more responsive to user interaction than previous versions of HTML. DHTML is a set of technologies that is used to enhance the functionalities and appearance of a Web page. Simple examples of DHTML pages would include (1) having the color of a text heading change when a user passes a mouse over it or (2) allowing a user to "drag and drop" an image to another place on a Web page. Dynamic HTML can allow Web documents to look and act like desktop applications or multimedia productions.

The biggest obstacle to the use of dynamic HTML is that, since many users are still using older browsers, a Web site must create two versions of each site and serve the pages appropriate to each user's browser version due to its compatibility with Navigator 4.0 (part of Netscape's Communicator suite), and by Microsoft's browser, Internet Explorer 4.0 and above only.

Microsoft DHTML includes: HTML, JavaScript/Jscript, Cascading Style Sheet, the Dynamic HTML Object Model and Event Model, ActiveX Controls and other related technologies.

2.2.1.3 XML

XML (Extensible Markup Language) is a flexible way to create common information formats and share both the format and the data on the World Wide Web and intranets. For example, computer makers might agree on a standard or common way to describe the information about a computer product (processor speed, memory size, and so forth) and then describe the product information format with XML. Such a standard way of describing data would enable a user to send an intelligent agent (a program) to each computer maker's Web site, gather data, and then make a valid comparison. XML can be used by any individual or group of individuals or companies that wants to share information in a consistent way.

Both XML and HTML contain markup symbols to describe the contents of a page or file. HTML, however, describes the content of a Web page (mainly text and graphic images) only in terms of how it is to be displayed and interacted with. XML describes the content in terms of what data is being described. For example, the word "phonenum" placed within markup tags could indicate that the data that followed was a phone number. This means that an XML file can be processed purely as data by a program or it can be stored with similar data on another computer or, like an HTML file, that it can be displayed. For example, depending on how the application in the receiving computer wanted to handle the phone number, it could be stored, displayed, or dialed.

The markup symbols for XML are unlimited and self-defining. Applications related to banking, e-commerce ordering, personal preference profiles, purchase orders, litigation documents, part lists, and many others are anticipated.

2.2.2 *Scripting Languages*

2.2.2.1 *JavaScript/Jscript*

JavaScript is an interpreted programming or script language from Netscape. In general, script languages are easier and faster to code. Script languages generally take longer to process than compiled languages, but are very useful for shorter programs.

JavaScript is used in Web site development to do such things as:

- Automatically change a formatted date on a Web page.
- Cause a linked-to page to appear in a popup window.
- Cause text or a graphic image to change during a mouse rollover.

JavaScript code can be imbedded in HTML pages and interpreted by the Web. JavaScript can also be run at the server as in Microsoft's Active Server Pages (Active Server Page) before the page is sent to the requestor. Both Microsoft and Netscape browsers support JavaScript, but sometimes in slightly different ways.

2.2.2.2 VBScript

VBScript is an interpreted script language from Microsoft that is a subset of its Visual Basic programming language. VBScript can be compared to other script languages designed for the Web, including:

- Netscape's JavaScript
- Sun Microsystem's Tool Command Language
- The UNIX-derived Practical Extraction and Reporting Language
- IBM's Restructured Extended Executor

In general, script languages are ideal for smaller programs of limited capability or that can reuse and tie together existing compiled programs.

VBScript is Microsoft's answer to Netscape's popular JavaScript. Both are designed to work with an interpreter that comes with a Web browser - that is, at the user or client end of the Web client/server session. VBScript is designed for use with Microsoft's Internet Explorer browser together with other programming that can be run at the client, including ActiveX control, automation servers, and Java applet. Although Microsoft does support Netscape's JavaScript (it converts it into its own JScript), Netscape does not support VBScript. For this reason, VBScript is best used for intranet Web sites that use the Internet Explorer browser only.

2.2.3 *Server Side Software Development*

2.2.3.1 ASP – Active Server Page

An Active Server Page (ASP) is an HTML page that includes one or more script (small embedded programs) that are processed on a Microsoft Web server before the page is sent to the user. An ASP is somewhat similar to a Server-side include or a common gateway interface (CGI) application in that all involve programs that run on the server, usually tailoring a page for the user. Typically, the script in the Web page at the server uses input received as the result of the user's request for the page to access data from a database and then builds or customizes the page on the fly before sending it to the requestor.

ASP is a feature of the Microsoft Internet Information Server (IIS), but, since the server-side script is just building a regular HTML page, it can be delivered to almost any browser. You can create an ASP file by including a script written in VBScript or JScript in an HTML file or by using ActiveX Data Objects (ADOs) program statements in the HTML file. You name the HTML file with the ".asp" file suffix. The server-side script will result in an easily displayable HTML page. Client-side scripts (for example, with JavaScript) may not work as intended on older browsers.

2.2.3.2 JSP – Java Server Page

Java Server Page (JSP) is a technology for controlling the content or appearance of Web pages through the use of servlet, small programs that are specified in the Web page and run on the Web server to modify the Web page before it is sent to the user who requested it. Sun Microsystems, the developer of Java, also refers to the JSP technology as the Servlet application program interface (API). JSP is comparable to Microsoft's Active Server Page (ASP) technology. Whereas a Java Server Page calls a Java program that is executed by the Web server, an Active Server Page contains a script that is interpreted by a script interpreter (such as VBScript or JScript) before the page is sent to the user.

An HTML page that contains a link to a Java servlet is sometimes given the file name suffix of .JSP.

2.2.3.3 Servlet

A servlet is a small program that runs on a server. The term was coined in the context of the Java applet, a small program that is sent as a separate file along with a Web (HTML) page. Java applets, usually intended for running on a client, can result in such services as performing a calculation for a user or positioning an image based on user interaction.

Some programs, often those that access databases based on user input, need to be on the server. Typically, these have been implemented using a Common Gateway Interface (CGI) application. However, with a Java running in the server, such programs can be implemented with the Java programming language. The advantage of a Java servlet on servers with lots of traffic is that they can execute more quickly than CGI applications. Rather than causing a separate program process to be created, each user request is invoked as a thread in a single daemon process, meaning that the amount of system overhead for each request is slight.

Instead of a URL that designates the name of a CGI application (in a "cgi-bin" subdirectory), a request in a form on a Web HTML page that results in a Java servlet getting called would call a URL that looks like this:

http://www.whatis.com:8080/servlet/gotoUrl?http://www.someplace.com

The "8080" port number in the URL means the request is intended directly for the Web server itself. The "servlet" would indicate to the Web server that a servlet was being requested.

Add-on modules allow Java servlets to run in Netscape Enterprise, Microsoft Internet Information Server (IIS), and Apache servers [5].

2.2.4 Web Server

A Web server is a computer that is designed to act as a hub for a number of computers that form a network, it is a network server that manages access to files, folders and other resources over the Internet or a local Intranet via platform-neutral HTTP (Hyper Text Transfer Protocol). In addition, Web servers posses Web networking characteristics. They handle permissions, execute programs, keep track of directories and files and communicate with client computer. These client computers make requests for files and actions from server computers using HTTP, an Internet protocol that enables the distribution of hypertext documents.

If individual networks are analogous to localized systems of roads, then their Web servers are the traffic lights at the main intersections. Web servers handle all the incoming and outgoing data of the network, and make sure that everything that arrives at the Web server is sent on the right road to reach its destination. The Internet is, in essence, a network connecting many smaller networks (the "information superhighway"). All information accessible on some servers, and when a client requests to view the information, the client are actually accessing the file on a server somewhere in the world. When information is uploaded to the Internet, it is also placed on a server.

To set up a computer to function as a Web server, special Web server software as follow can be used.

2.2.4.1 IIS – Microsoft Internet Information Server

IIS (Internet Information Server) is a group of Internet server (Web or Hypertext Transfer Protocol, File Transfer Protocol, and Gopher) and other capabilities for Microsoft's Windows NT and Windows 2000 Server operating systems. IIS is Microsoft's bid to dominate the Internet server market that is also addressed by Netscape, Sun Microsystems, O'Reilly, and others. With IIS, Microsoft includes a set of programs for building and administering Web sites, a search engine, and support for writing Web-based applications that access database. Microsoft points out that IIS is tightly integration with the Windows NT and 2000 Servers in a number of ways, resulting in faster Web page serving.

A typical company that buys IIS can create pages for Web sites using Microsoft's Front Page product (with its WYSIWYG user interface). Web developers can use Microsoft's Active Server Page (Active Server Page) technology, which means that applications - including ActiveX control controls - can be imbedded in Web pages that modify the content sent back to users. Developers can also write programs that filter requests and get the correct Web pages for different users by using Microsoft's Internet Server Application Program Interface interface. ASPs and ISAPI programs run more efficiently than Common Gateway Interface (common gateway interface) and server-side include (Server-side include) programs, two current technologies.

Microsoft includes special capabilities for server administrators designed to appeal to Internet service providers (Internet service provider). It includes a single window (or "console") from which all services and users can be administered. It's designed to be easy to add components as "snap-ins" that you didn't initially install. The administrative windows can be customized for access by individual customers.

IIS includes security features and promises that it is easy to install. It works closely with the Microsoft Microsoft Transaction Server to access databases and provide control at the transaction level. It also works with Microsoft's Netshow in the delivery of streaming audio and video, delayed or live.

2.2.4.2 PWS Microsoft Personal Web Server

PWS, an abbreviation for Personal Web Server, is Microsoft's version of a Web server program for individual PC users who want to share Web pages and other files from their hard drive. PWS is a scaled-down version of Microsoft's more robust Web server, Internet Information Server IIS. PWS can be used with a full-time Internet connection to serve Web pages for a Web site with limited traffic. It can also be used for testing a Web site offline or from a "staging" site before putting it on a main Web site that is exposed to larger traffic.

PWS can be used together with Microsoft's FrontPage, a Web site design product, to upload Web pages from a remote location or to the local hard drive; to check for dead links; to create directories; and to set permissions. PWS is frequently used as part of the trend toward peer-to-peer exchange and publishing.

2.2.4.3 Apache Web Server

Apache is a freely available Web server that is distributed under an "open source" license. Version 1.3 runs on most UNIX-based operating systems (such as Linux, Solaris, Digital UNIX, and AIX), on other UNIX/POSIX-derived systems (such as Rhapsody, BeOS, and BS2000/OSD), on AmigaOS, and on Windows NT/95/98. According to the Netcraft (www.netcraft.com) Web server survey in September 1998, more than 50% of all Internet servers were running Apache. Although Windows-based systems with Web servers from Microsoft, Netscape, and other companies are probably gaining in terms of numbers, Apache is likely to remain popular in enterprises and server locations (such as universities) where UNIX-based systems are prevalent.

Apache complies with the newest level of the Hypertext Transport Protocol, HTTP 1.1. Free support is provided through a bug reporting system and several Usenet newsgroups. Several companies offer priced support.

2.2.4.4 W3C Jigsaw Web Server

W3C (World Wide Web Consortium) is a free Web server written entirely in Java. Two of its major design goals are portability and extensibility. Jigsaw runs on most machines that support Java environment, Microsoft Windows series, Sun Microsystem's Solaris and Linux.

Writing the Web server with Java makes it an object-oriented Web server and has extends Jigsaw server's functionality. It would be able to replace CGI scripts. The major components are the daemon module and the resource module. The daemon module deals with HTTP. It handles incoming connections, creates new client objects, decodes requests and sends replies. The resource module is responsible for managing the information space on the server.

2.2.5 Java

2.2.5.1 Java in Intelligent Agent

The commercialization of the Internet has continued unabated. Java has proved that it is the language of choice for constructing Web site, creating an application that runs on server, embedded into Web pages etc. Java also promise "write once, run anywhere" where it is portable to any platforms. An aspect of "architecture-neutral" makes it ideal for programming the Internet. It allows a user to receive software from a remote system and execute it on a local system, regardless of the underlying hardware or operating system. An interpreter and runtime called Java Virtual Machine (JVM) insulates the software from the underlying hardware.

Java code does not translated into the machine instructions for a particular computer platform. Instead, Java source code (.java) is compiled into an intermediate form called "bytecode" which are stored in a class file. These bytecode can be executed on any computer platform that implements a JVM. The language is very powerful in term of portability and development of the cross-platform applications.

Because of its portability, it allows the bytecode being transferred across the network. Java has been an ideal language in cross-platform application development though it does raised up the question of security on the system with its mobility. The performance of executing time for Java is obviously slower than those traditional programming languages because languages like C and C++ compile programming code straight into machine language rather that bytecode which needs an interpreter. The solution to this is the introduction of Just-In-Time compiler (JIT). It runs concurrently with JVM. It monitors and determines the codes which are the most frequent used in the machine and compiles the particular codes into machine instruction which will boost up the overall performance.

2.2.5.2 Multithreading

It would be nice if we could "do one thing at a time". Concurrency is essential in every aspect. It will be great if a programming language allows concurrency but generally provide only a simple set of control structures that enable programmers to perform one action at a time and then proceed to the next action after the previous one is finished [6].

Java is unique among popular general-purpose programming language in that is makes concurrency primitives available to the applications programmer. The programmer specifies that applications contain threads of execution, each thread designation a portion of a program that may execute concurrently with other threads. This capability is called multithreading. It gives the Java programmer a powerful capabilities not available in C and C++, the language on which Java is based. C and C++ are called single-threaded languages.

An example of how multithreading works is when programs download large files such as audio clips or video clips from the World Wide Web, we do not want to wait until an entire clip is downloaded before starting the playback. So we can put multiple threads to work: one that downloads a clip, and another that plays the clip so that these activities, or tasks, may proceed concurrently. To avoid choppy playback, we will coordinate the threads so that the player thread does not begin until there is a sufficient amount of the clip in memory to keep the player thread busy. Another example of multithreading is Java's automatic garbage collection. C and C++ place with the programmer the responsibility for reclaiming dynamically allocated memory that is no longer needed.

2.2.5.3 JavaBeans

JavaBeans is an object-oriented programming interface from Sun Microsystems that lets you build re-useable applications or program building blocks called component that can be deploy in a network on any major operating system platform [6]. Like Java applet, JavaBeans components (or "Bean") can be used to give World Wide Web pages (or other applications) interactive capabilities such as computing interest rates or varying page content based on user or browser characteristics. From a user's point-of-view, a component can be a button that you interact with or a small calculating program that gets initiated when you press the button. From a developer's point-of-view, the button component and the calculator component are created separately and can then be used together or in different combinations with other components in different applications or situations. When the components or Beans are in use, the properties of a Bean (for example, the background color of a window) are visible to other Beans and Beans that haven't "met" before can learn each other's properties dynamically and interact accordingly.

Beans are developed with a Beans Development Kit (BDK) from Sun and can be run on any major operating system platform (Windows 95, UNIX, Mac) inside a number of application environments (known as containers), including browsers, word processors, and other applications. To build a component with JavaBeans, you write language statements using Sun's Java programming language and include JavaBeans statements that describe component properties such as user interface characteristics and events that trigger a bean to communicate with other beans in the same container or elsewhere in the network.

Beans also have persistence, which is a mechanism for storing the state of a component in a safe place. This would allow, for example, a component (bean) to "remember" data that a particular user had already entered in an earlier user session. JavaBeans gives Java applications the compound document capability that the OpenDoc and ActiveX interfaces already provide.

2.2.5 Configuring Mail Server

2.2.4.5 IMAP - Internet Message Access Protocol

Internet Message Access Protocol (IMAP) is a standard protocol for accessing e-mail from your local server. IMAP (the latest version is IMAP4) is a client/server protocol in which e-mail is received and held for you by your Internet server. You (or your e-mail client) can view just the heading and the sender of the letter and then decide whether to download the mail. You can also create and manipulate folders or mailboxes on the server, delete messages, or search for certain parts or an entire note. IMAP requires continual access to the server during the time that you are working with your mail [7].

A less sophisticated protocol is Post Office Protocol 3 (POP3). With POP3, your mail is saved for you in your mail box on the server. When you read your mail, all of it is immediately downloaded to your computer and no longer maintained on the server.

IMAP can be thought of as a remote file server. POP can be thought of as a "store-and-forward" service.

POP and IMAP deal with the receiving of e-mail from your local server and are not to be confused with Simple Mail Transfer Protocol (SMTP), a protocol for transferring e-mail between points on the Internet. You send e-mail with SMTP and a mail handler receives it on your recipient's behalf. Then the mail is read using POP or IMAP.

2.2.5.2 POP3 - Post Office Protocol 3

POP3 (Post Office Protocol 3) is the most recent version of a standard protocol for receiving e-mail. POP3 is a client/server protocol in which e-mail is received and held for you by your Internet server. Periodically, you (or your client e-mail receiver) check your mail-box on the server and download any mail.

An alternative protocol is Internet Message Access Protocol (IMAP). With IMAP, you view your e-mail at the server as though it was on your client computer. An e-mail message deleted locally is still on the server. E-mail can be kept on and searched at the server. POP can be thought of as a "store-and-forward" service. IMAP can be thought of as a remote file server.

POP and IMAP deal with the receiving of e-mail and are not to be confused with the Simple Mail Transfer Protocol (SMTP), a protocol for transferring e-mail across the Internet. You send e-mail with SMTP and a mail handler receives it on your recipient's behalf. Then the mail is read using POP or IMAP.

2.2.5.3 SMTP - Simple Mail Transfer Protocol

SMTP (Simple Mail Transfer Protocol) is a TCP/IP protocol used in sending and receiving e-mail. However, since it's limited in its ability to queue messages at the receiving end, it's usually used with one of two other protocols, POP3 or Internet Message Access Protocol, that let the user save messages in a server mailbox and download them periodically from the server. In other words, users typically use a program that uses SMTP for sending e-mail and either POP3 or IMAP for receiving messages that have been received for them at their local server. Most mail programs such as Eudora let you specify both an SMTP server and a POP server. On UNIX-based systems, sendmail is the most widely-used SMTP server for e-mail. A commercial package, Sendmail, includes a POP3 server and also comes in a version for Windows NT.

2.3 DATABASE

2.3.1 DBMS – Database Management System

2.3.1.1 Microsoft Access

The latest version is Access 2000. It is a relational database management system (RDBMS) designed for small companies and personal use. It utilizes Microsoft Jet Engine. The program is very user-friendly and has simple application creation and report generating tools.

2.3.1.2 Microsoft SQL Server

The most current version is version 7.0. It is a scalable, high performance database management system designed specifically for distributed client/server application. It provides tight integration with Windows and Windows-based applications. It is ideal for powering the Web sites too. Through tight integration with IIS, SQL Server can be queried and updated via popular Web browsers.

2.3.1.3 ODBC - Open Database Connectivity

Open Database Connectivity (ODBC) is an open standard application programming interface (API) for accessing a database. By using ODBC statements in a program, you can access files in a number of different databases, including Access, dBase, DB2, Excel, and Text. In addition to the ODBC software, a separate module or driver is needed for each database to be accessed. The main proponent and supplier of ODBC programming support is Microsoft.

ODBC is based on and closely aligned with The Open Group standard Structured Query Language (SQL) Call-Level Interface. It allows programs to use SQL requests that will access databases without having to know the proprietary interfaces to the databases. ODBC handles the SQL request and converts it into a request the individual database system understands.

ODBC was created by the SQL Access Group and first released in September, 1992. Although Microsoft Windows was the first to provide an ODBC product, versions now exist for UNIX, OS/2, and Macintosh platforms as well.

In the newer distributed object architecture called Common Object Request Broker Architecture (CORBA), the Persistent Object Service (POS) is a superset of both the Call-Level Interface and ODBC. When writing programs in the Java language and using the Java Database Connectivity (JDBC) application program interface, you can use a product that includes a JDBC-ODBC "bridge" program to reach ODBC-accessible databases.

2.3.1.4 JDBC - Java Database Connectivity

JDBC (Java Database Connectivity) is an application program interface (application program interface) specification for connecting programs written in Java to the data in popular database. The application program interface lets you encode access request statements in structured query language (Structured Query Language) that are then passed to the program that manages the database. It returns the results through a similar interface. JDBC is very similar to the SQL Access Group's Open Database Connectivity (Open Database Connectivity) and, with a small "bridge" program, you can use the JDBC interface to access databases through the ODBC interface. For example, you could write a program designed to access many popular database products on a number of operating system platform. When accessing a database on a PC running Microsoft's Windows 95 and, for example, a Microsoft Access database, your program with JDBC statements would be able to access the Microsoft Access database.

JDBC actually has two levels of interface. In addition to the main interface, there is also an API from a JDBC "manager" that in turn communicates with individual database product "driver", the JDBC-ODBC bridge if necessary, and a JDBC network driver when the Java program is running in a network environment (that is, accessing a remote database).

When accessing a remote database, JDBC takes advantage of the Internet's file addressing scheme and a file name looks much like a Web page address (or Uniform Resource Locator). For example, a Java SQL statement might identify the database as:

jdbc:odbc://www.somecompany.com:400/databasefile

JDBC specifies a set of object-oriented programming class for the programmer to use in building SQL requests. An additional set of classes describes the JDBC driver API. The most common SQL data type, mapped to Java data types, are supported. The API provides for implementation-specific support for Microsoft Transaction Server requests and the ability to commit or roll back to the beginning of a transaction.

2.4 ANALISIS ON OTHER SPORT AGENT

2.4.1 Review of Sports Sleuth

SportsSLEUTH is an Internet sports information agent. By passing some parameters, users can get the information about they favorite teams in NBL (baseball), NBA (basketball), NFL (American football) and NHL (American ice hockey) from United State via email. With the SportsSLEUTH expertise on users' side, users can stakeout and track the teams they choose, getting daily reports, via e-mail, on their local and national news, player transactions, statistics, injuries, message boards, and much, much more!
Once users select the teams they would like SportsSLEUTH to stakeout, they will receive a daily e-mail containing team information covering the following areas [8]:

- National News (from top sources like ESPN, CNN/SI, USA Today, Sporting News, Yahoo, Fox, etc.)
- Local News (from hundreds of local and regional sources from around the nation)
- Live Scores (up-to-the-second scores as the action happens!)
- Message Boards (from reputable sources like ESPN and Sports Only)
- Transactions
- Injury Reports
- Newsgroups
- Previews
- Final Results
- Game Recaps
- Schedules
- Odds
- Standings
- Game Day Weather Reports

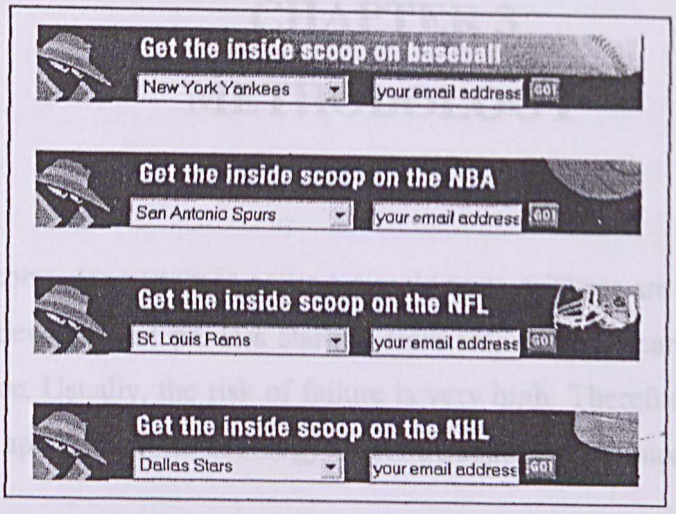


Figure 2-8: The interface of the SportsSLEUTH agent.

2.4.2 Comments

The SportsSLEUTH agent provides variety of useful information to their subscribers in form of email. But since the agent will send mail to us, we still have to get ourselves connected to the Internet to view the summarize reports. The most importantly, the agents just won't suit the "culture" in Malaysia where merely small number of Malaysians love these American games. Furthermore, the service is not covering the soccer news especially English Premier League. The agent won't be able to keep informing the users about the users about the latest news via SMS which the user need not have to connect themselves to the Web to know the score instead of letting the agent reports the latest score on their own in the server LIVE!

Therefore, the planned agent is on the exact purpose to fulfill the "weakness" of the current sport agent though the SportsSleuth does very powerful in gathering information about the sport but it does not provide what Malaysians soccer fans needed.

CHAPTER 3

METHODOLOGY

Developing a computer system is never a simple matter. There are many facets involved in the system development process starting from information search to implementation and maintenance. Usually, the risk of failure is very high. Therefore, proper planning of the project and appropriate methodology must be adopted to produce a good outcome.

This chapter will draw out the suitable methodology for the proposed system, the stages involved and appropriate software tools.

3.1 WATERFALL MODEL WITH PROTOTYPING

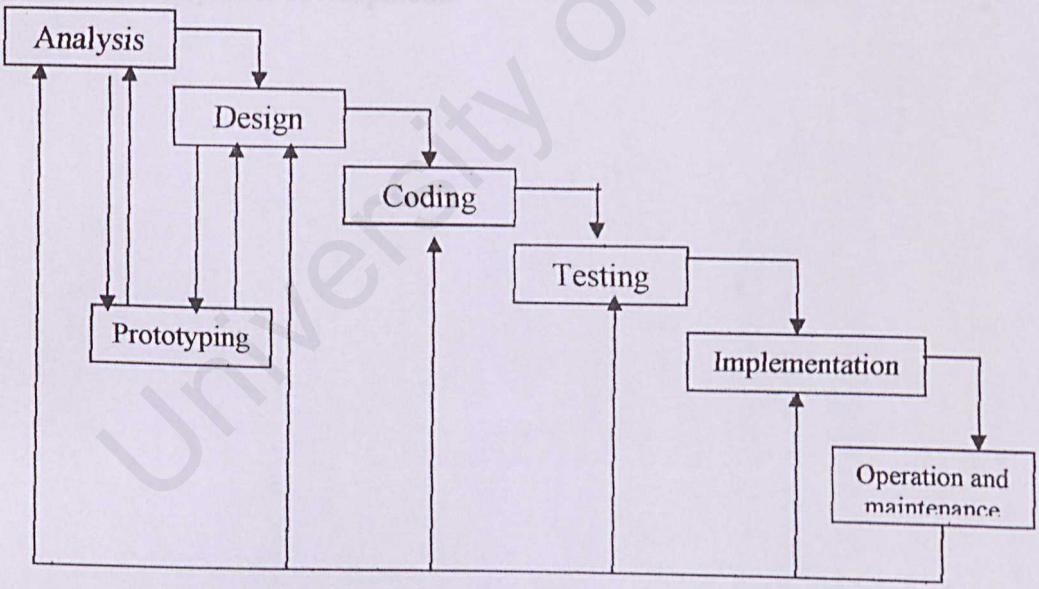


Figure 3-1: Waterfall Model With Prototyping

3.1.1 Methodology consideration

Waterfall Model was the earliest model and many of the new approaches are modified from this model. When the Waterfall model had been designed, it received many critics and influence but it can be improve by implementing the model with prototyping. Thus, the new model becomes more suitable and reliable during system development [9].

With prototyping is implemented into the Waterfall model, it solves a lot of the problems cause by the traditional model, which a software development will evolves lots of iteration processes as listed from the traditional Waterfall model. A prototype is a partially developed product that enables customers and developer to test the proposed system and decide if it is suitable for the final product.

This model has gathered the advantages of both traditional Waterfall model and Prototyping model. Hence to improve the quality of the software life circle process and guarantee the quality of the final delivery system. This is the main reason the model is implemented for this system development.

3.1.2 Prototyping solves the problems of traditional Waterfall Model

The ability of letting developers to quickly create a prototype to verify the needs particular process has been one of the most powerful features this model has. Therefore, the revisions are made at the requirements stage rather than the more costly testing stage, as we understand that a software development process may involve a lot of iterations process. A good example is when the developers have to use the 'try and error' method to get the best result; and if the result is not feasible or failed to hit the target, they will have to start all over from the beginning of the process again and again. Thus the prototyping stage is used to examine some aspects of the proposed system for the first few stages. The model will organize a systematical way to manage the development process and beware of using an inappropriate method for the system development. By validation of these stages, it will effectively reduce the possibility of repeating any process caused by the using of any inappropriate methods. Therefore, when the software development process come to the system testing stage, it will automatically validate the requirements of the system and also verify the system design as planning in the earlier stages.

	Prototyping Model	Waterfall Model
Advantages	▪ Reduce risk and uncertainty in development	▪ It is easier for the explanation to a customer who are not familiar with software development. This model will software developer during the developing process. ▪ Most of the latest models are built or modified according to this model.
Disadvantages	▪ Produce the product in limited time will ignore the quality for the software. Thus, more time is needed to maintain the system. ▪ Developers may develop a system within unsuitable platforms or programs.	▪ Not showing how the basic coding is designed or created. ▪ Do not have any reference to refer to when any disaster or changes is happening to the product or activities ▪ Failed to face the software as the problems solving process, which we notice that the waterfall model is actually modified from hardware development process.

Table 3-2: The comparison between the Prototyping Model and Waterfall Model.

The waterfall model with prototyping approach that will be adapted in the proposed project encompasses the activities at system analysis, system design, coding, testing and implementation. Each of the stage is discussed below.

Analysis

The main activity at this stage is to understand the proposed system and determine the system requirement. This will involve data gathering and system analysis. Other task is observing other intelligent agent software (especially some intelligent agent which has the similarity behavior or function like the proposed system) with the intention to mimic or enhance the current system. Thus, the accuracy of the proposed system is specified correctly. Furthermore, many AI reasoning and inference techniques have been studied to determine the finest and most suitable technique to apply into this project.

Design

This stage concerns about the front-end design, database design and system design of the intelligent agent. Data Flow Diagram (DFD) modeling will be involved in the system design while the logical design of the Microsoft Access database will be depicted in a proposed database structure design.

Coding

This stage translates and implements the detail design representation of the system into programming realization. Java will be used in the most of the programming process including the coding of the rule base, which will be inferenced to parse and search for some keywords from HTML code. Scripting languages such as VBScript and HTML are used in coding together with ASP during handling requests from the users via Internet. Microsoft FrontPage is the proposed Web-authoring tool that will be used to create Web pages while Microsoft Access 2000 will be used to develop the database of the system.

Testing

Testing is a critical step in assuring the quality of the developed intelligent agent and will represent the ultimate review of specification, design and coding. First, unit testing will be performed to verify each program module. The testing on the rule package will be particularly difficult because there will be a lot of unexpected result produced. Next, integration testing is performed. It is to integrate unit-tested program modules and conduct tests that uncover errors or bugs associated with the interfacing of those modules. Validation test succeeds when the system functions in the manner that is reasonably expected.

Implementation

Implementation is done during the end of the system development life cycle. The system will deploy into the target environment. In this case, Web server.

Operation and Maintenance

It is an ongoing process throughout the system's lifetime. Necessary adjustments such as new rules or better inference technique might be carried out. All these will ensure a good system during its operation on the Web server.

3.2 DEVELOPMENT ENVIRONMENT

3.2.1 Operating System

Windows 2000 Professional

Windows 2000 Professional is the Windows operating system for business desktop and laptop systems. It is used to run software applications, connect to Internet and intranet sites, and access files, printers, and network resources.

Windows 2000 Professional is built on Windows NT technology and the easy-to-use, familiar Windows 98 user interface. It gives business users increased flexibility. Microsoft had take full advantage of the new technologies in Windows 2000, delivering products that are more manageable and reliable, thereby reducing the cost of computing. Working in partnership with hundreds of Independent Software Vendors, (ISVs) Microsoft has developed a directory of Windows 2000 compatible applications. This directory lists products that have been tested on the Windows 2000 platform and are ready for deployment [10].

Advantages of Windows 2000 Professional

a) *PCs stand up and running*

Windows 2000 includes a built-in safeguard called Windows File Protection. This feature helps prevent critical operating system files from being deleted or altered by users or applications.

If a system file should be changed or deleted, Windows File Protection can detect the change, retrieve a correct version of the file from a cache, and restore it to the system file folder. The end user never knows the repairs have been made because Windows 2000 just keeps running.

b) *Protects Against User Error*

Microsoft provides "self healing applications" to repair the mistake made by user by incorrectly installed or removed an application, or accidentally changed one of the application files, which could cause a system failure.

c) *Fewer Reboots*

Performing routine maintenance on your system requires significantly fewer reboots, therefore less downtime, with Windows 2000. In addition, with its support for Plug and Play, Windows 2000 automatically recognizes and adapts to hardware changes.

Windows 2000 Professional is up to 30 percent faster and, according to Microsoft and Independent Hardware and Software Testing (NTSL) test, 13 times more reliable than Windows 98. In short, Windows 2000 Professional is the most reliable Windows ever.

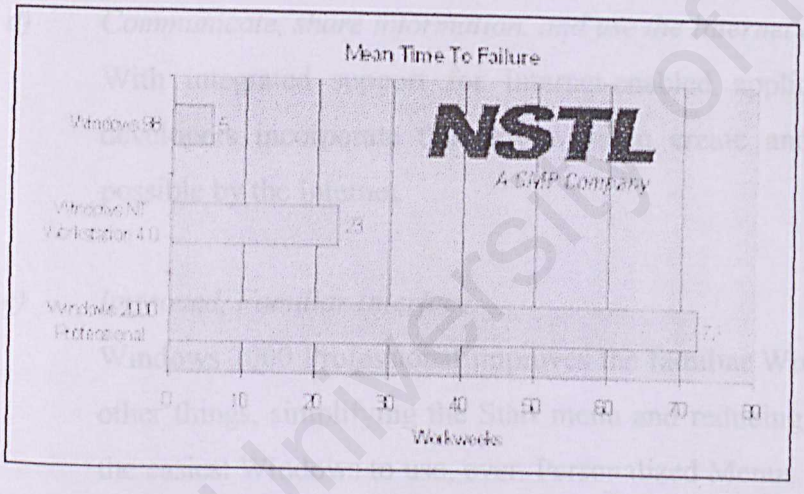


Figure 3-3: Windows 2000 mean time to failure greatly exceeds that of Windows NT Workstation 4.0 and Windows 98.

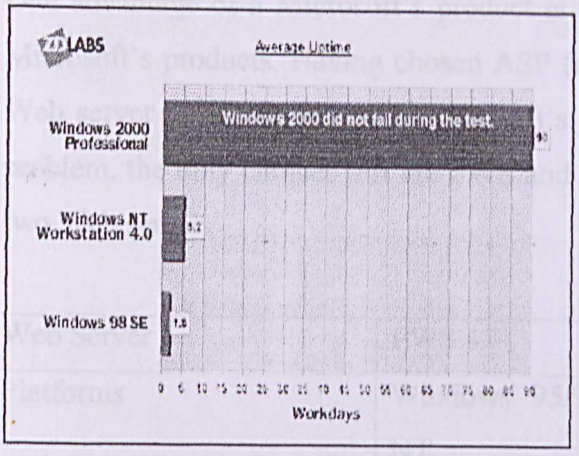


Figure 3-4: Windows 2000 Professional simply did not fail during the 90 days it was tested by ZD Labs.

d) *Communicate, share information, and use the Internet quickly and easily*

With integrated support for Internet-enabled applications, business software developers incorporate the new ways to create and share information made possible by the Internet.

e) *Improved, Familiar Interface*

Windows 2000 Professional improves the familiar Windows interface by, among other things, simplifying the Start menu and reducing desktop clutter, making it the easiest Windows to use, ever. Personalized Menus, a new "smart" feature that adapts the Start menu to the way you work, only displays the applications you use most often.

f) *Most Secure Windows*

Windows 2000 Professional incorporates the security system that is a part of every Windows NT operating system. It allows users and administrators to select the appropriate level of protection for their information and applications.

3.2.2 *Web Server*

One advantage of a Microsoft's product is the ability to work seamlessly among other Microsoft's products. Having chosen ASP (discuss at the later section), the choices for Web server will naturally be a Microsoft's product too. To eliminate the compatibility problem, the only choices left are PWS and IIS. Following is the caparison between the two Web server.

Web Server	PWS 4.0	IIS 4.0
Platforms	Windows 95/98, Windows NT	Windows NT, Windows 2000
Brief Description	Support up to 10 concurrent connections; no longer support FTP	A most popular Web server for windows platforms.
Features	A great entry-level Web server for publishing your own Web pages.	Updated versions of IIS, Index Server, Message Queue Server (MSMQ) and Transaction Server (MTS).

Table 3-5: The comparison between PWS 4.0 and IIS 4.0.

Internet Information Server 4.0 (IIS)

From the table above, IIS is chosen because of the additional facilities provided. IIS is an enterprise level Web server software while PWS is only at entry level. Internet Information Server 4.0 is a high performance Web application server for Windows 2000 Professional. It brings many advanced capabilities to Web professionals, both as a Web server for corporate intranets and public Internet sites as well as a platform for the next generation of line-of-business applications.

This Web server also incorporates World Wide Web (WWW), File Transfer Protocol (FTP), Microsoft Index Server and Secure Socket Layer (SSL) services. Furthermore, it also provides comprehensive set of tools for managing the Web server and its components [11].

With IIS, it is an easy task to build scalable and reliable application for the Web. Developer can easily integrate the benefits of transactions into Web applications using IIS. IIS also brings together the advantages of Windows 2000 Professional, with a range of services for client/server development and the standards of the Internet to create a true Web platform for distributed application.

3.2.3 Web Application Programming Languages

VBScript

VBScript is an interpreted script language from Microsoft that is a subset of its Visual Basic programming language. In general, script languages are easier and faster to code in than the more structured, compiled languages such as C and C++ and are ideal for smaller programs of limited capability or that can reuse and tie together existing compiled programs. The function of VBScript is to make Web application more dynamic. VBScript is chosen because it is more reliable to work with ASP.

It is implemented as a fast, portable, lightweight interpreter for use in WWW browsers and other application that use Microsoft ActiveX controls, Automation Servers and Java applets. VBScript currently is available as a part of Internet Explorer and IIS [12].

When used with Internet Explorer, VBScript is directly comparable to JavaScript. Like JavaScript, VBScript is a pure interpreter that processes source code embedded directly in the HTML. VBScript code, like JavaScript does not produces standalone applets but is used to add intelligence and interactivity to HTML document. For programmers who already know Microsoft Visual Basic, VBScript is a valuable alternative to JavaScript in developing Web pages.

JavaScript

JavaScript is an interpreted programming or script language from Netscape. It is used in Web site development to do such things as: Automatically change a formatted date on a Web page, cause a linked-to page to appear in a popup window, cause text or a graphic image to change during a mouse rollover. JavaScript is useful for adding interactivity to the World Wide Web because scripts can be embedded in HTML files. JavaScript can also be run at the server as in Microsoft's Active Server Pages before the page is sent to the requestor.

Active Server Pages (ASP)

Evaluation Criteria	CGI	ASP	Servlets
Rapid Development	Good	Very Good	Good
Supportability	Average	Very Good	Good
Database Connectivity	Very Good	Very Good	Very Good
Interactivity	Good	Good	Very Good

Table 3-6: The comparison between the three Web servers.

From the table above, ASP is chosen as the server-side programming paradigm. ASP is a script that runs in Microsoft Web Server. It is used to generate HTML scripts for the client browser. Compared to CGI, ASP is easier to use and is more flexible in changing code as no compilation is involved and so it is chosen as the main development tool for server run script.

Active Server Pages (ASP) is the latest server-based technology from Microsoft for building dynamic and interactive Web pages. The basic of ASP is the Microsoft IIS software.

ASP is not an application. It is a VBScript and JavaScript interpreter that is integrated with IIS, together with an interface for other custom component. It is also able to include other Web pages components like ActiveX controls and Java Applets. Therefore, ASP is considered as a glue technology, which binds together other various server-based system to help build interactive Web pages [13].

ASP is used to build Web pages because:

- It is suitable for publishing and collecting data on the Web.
- It provides way for building secure transactions, server-based applications and Web sites.
- ASP works together with Windows NT and IIS to provide a comprehensive set of key software technology, which enable secure exchange of information over public network, access control to server resources and confident identification of server and client.
- Furthermore, ASP also provides ADO, one of ASP component that allows easy but powerful connections to be made to almost any database system for which an ODBC driver is available.
- It has pre-build ASP components, which provide plug-in object that perform specific tasks.
- Besides this, ASP can interact with almost any existing dynamic Web page technology such as CGI, ISAPI and scripts written in PERL, Python and Awk.
- It also suitable for building multi-tier Internet and Intranet applications.
- Supports client-server programming. Furthermore, the combination of ASP, client-side scripting and object can be used to create client-server applications.
- With ASP, it is able to create client-side code dynamically on server.

Hypertext Markup Language (HTML)

To publish information for global distribution, one needs a universally understandable language, just like a kind of publishing mother tongue that all computers may potentially understand. The publishing language used by the WWW is HTML.

It is a basic tool for that is needed for the development of client browser run script. HTML is the set of "markup" symbols or codes inserted in a file intended for display on a

World Wide Web browser. The markup tells the Web browser how to display a Web page's words and images for the user. The individual markup codes are referred to as elements.

HTML gives author the means to do the following:

- Publish online document with headings, text, tables, lists, photos and so on.
- Retrieve online information via hypertext links at the click of a button.
- Design form for conducting transactions with remote services, for use in searching for information and so on.
- HTML also includes spreadsheets, video clips, sound clips and other applications directly in its documents.

Most people agree that HTML document should work well across different browsers and platforms. Achieving interoperability lowers costs to content providers since they must develop only one version of a document. If the effort is not made, there is much greater risk that the Web will be devolved into a proprietary world of incompatible formats ultimately reducing the Web's commercial potential for all participants.

Each version of HTML has attempted to reflect greater consensus among industry players so that the investment made by content providers will not be wasted and that their document will not become unreadable in a short period of time. HTML has been developed with the vision that all manner devices should be able to use information on the Web. PC with graphics displays of varying solutions and color depths, cellular telephone, hand held devices, devices for speech for output and input, computers WITH high or low bandwidth and so on [14].

3.2.4 *Web Application Development Tools*

Microsoft Visual Interdev 2000

Microsoft Visual Interdev 2000 is the newest member of the visual tool family, is an integrated development tool for building dynamic Web applications accessible by any Web browser on any platform. It includes an integrated development environment, database connectivity tools, programmable components, site management and publishing capabilities, a personal Web server, content creation tools and many more.

Visual Interdev also includes a variety of development features for integrating client-server and Web technology. These features are enabled through Visual Interdev support for ActiveX controls, seamless database connectivity to any ODBC data source, support for building and testing large system and comprehensive support for the development of Web application.

Furthermore, it this tools also provides a rapid visual development environment for building ASP. It can easily integrate ActiveX server components written in Visual J++, Visual Basic, Visual FoxPro and Visual C++. Using Visual Interdev with ActiveX server components, a developer can easily creates multi-tier Web applications. ActiveX server component provides a convenient and effective way to tightly integrate a Web application with existing Internet system.

Visual Interdev delivers a comprehensive set of tightly integrated database tools for Web developers. The database connectivity features are based on the industry standard ODBC including Oracle, Microsoft SQL Server, Microsoft Access, Microsoft Visual FoxPro, Informix, Sybase, IBM DB/2 and so on. In addition, with Visual Interdev, a developer can create a scalable database solution because it leverages ASP. The core database components of Visual Interdev include ADO, Integrated Data View, Design-Time ActiveX Controls, Database Wizards, Query Designer and so on.

A Visual Interdev project consists of a live Web site. When developer opens a project, they are actually opening a live view of a Web site, as it exists on the Web server. The IDE is thus a complete Web site management tool that allows the developer to easily modify the structure of a Web site and to edit, add, move, rename and delete files and folders on the Web sites. It can also open multiple Web sites at the same time.

3.2.5 Database Management System

Microsoft Access 2000

It is used to serve as Database Management System for the system. Access is chosen because it can provide relational database power to manipulate information and is the easiest DBMS in managing database. It also supports SQL statements and can be easily integrated with programming languages used to develop the system. Since the database for the proposed system is not a complicated database, Microsoft Access is used because it is more suitable for regular database compare to Oracle, which is more suitable for building a large database.

3.2.6 *Inference Technique in Rule Base*

Forward chaining and backward chaining are two basic inference techniques used during reasoning and searching through the set of created rules. To determine which technique is the most suitable for the proposed system, hereby we discuss about the advantages for the both inference techniques on Table 3-9.

To identify the problem is the key to choose the proper and right approach. Referring to a book [15], it provides some very useful questions to be answered by the developers during the stage of determine which inference technique should be selected. The Table 3-10 unveils the questions.

From the Tables 3-9 and 3-10, the italic words are the important criteria to form an agent's inference way. Since there is a huge amount of HTML code in a Web page, gathering information and data collection during the parsing process are important for searching the right keywords. At the end of the process, a conclusion will be finalized where the team name, latest score or the player who had scored are identified. These identified strings of data will be sent later in the next stage.

As conclusion, forward chaining is the more suitable inference technique for the proposed system.

	Forward Chaining	Backward Chaining
Advantages	It works well when the problem naturally begins by *gathering information* and then seeing what can be inferred from it.	it works well when the problem naturally begins by forming a hypothesis and then seeing if it can be proven.
	Able to provide a considerable *amount of information* from only a small amount of data.	It remains focus only at one provided goal. Therefore, all the queries are derived from the goal and the queries will be very specific.
	It attempts to *infer everything* which is possible from the available information.	It searches only a particular part of the knowledge base that is relevant to the current problem.
	It is an excellent approach for certain types of problem solving tasks. Planning, *monitoring*, *interpretation*, and control.	It is an excellent approach for certain types of problem solving tasks. Diagnostics, prescription, and debugging.

Table 3-7: The advantages of the inference techniques.

	Forward Chaining	Backward Chaining
Questions If ...	Collecting data and then see what can be inferred from it?	Hypothesize a solution and then see if it can be proven?
In terms of the search space	More conclusions needed than data?	More data needed than conclusions?

Table 3-8: The questions to ask during decision making for the appropriate inference technique.

CHAPTER 4
SYSTEM ANALYSIS & DESIGN

This chapter will discuss about the design of the system. Each stage in the process has been disrupted into more details. Data Flow Diagram (DFD) will be used to describe the facets in the proposed system where each of the modules in the system has been drawn out. The details represented in this chapter will serve as a reference and important guidance for the system development phase as well as the system implementation and maintenance phase.

From the modules or sub modules below, the intelligence parts of the system will be seen clearly when they derived from ScoreAgent Module in the context level (Figure 4-3).

4.1 SYSTEM AND USER REQUIREMENTS

The system requirements need to be drawn out before develop a system. A requirement is a feature of the system or a description of the system is capable of doing in order to fulfill the system purpose [9]. There are two types of requirement, which is as followed:

- Functional requirement
- Non-functional requirement

4.1.1 Functional Requirement

A functional requirement describes an interaction between the system and its environment. It also describes how the system should behave when given a certain stimuli. The functional requirement for this system are stated below:

USER SECTION

a) Display Module

This is a Web pages front-end design, which are responsible for the interactions between user and the system. Users will pass the parameters or requests and the system will respond.

i) Registration and Login Form

Allow new user to register. Some particulars such as name and hand phone number will be needed. The information will be stored in database on server side. After that, the user can logins to the Web page and may proceeds their intention.

ii) Parameter Form

After validate user's login, they are allowed to change the parameter (the soccer team's name) base on their interest. Their request will be sent to ScoreAgent for attention. The agent will respond and sends the live score to user's mobile phone regarding to the team their had selected.

AGENT SECTION

a) Parsing and Monitoring Module

This is the "intelligence" module. The agent needs to parse the HTML text from the monitored Web site for some keywords such as the team's name or the updated score. The agent should be able to differentiate the changes on the score to determine whether the score has been updated before sends the parsed text to mail server.

Two packages from Java are used to send & retrieve information from the Web. There are *java.net.URL* and *java.net.HttpURLConnection*.

b) Mail Sending Module

A mail server software called Mail64 will be installed on the server. The server is designated the task to send the formatted parsed text to the user. The format to send e-mail is accountName@domainName. Therefore, to send a mail to telco's mail server will be handPhoneNumber@mailServerDomainName. For example, the agent will send to the address 0126599904@sms.maxis.net.my on other side of the mail server.

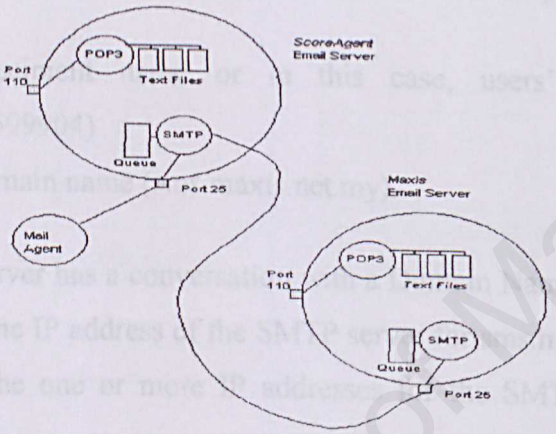


Figure 4-1: The conversations with other SMTP servers to deliver the e-mail.

This is the process of sending e-mail refers to the Figure 3-1 [7]:

- Agent connects to the SMTP server locally at mail.ScoreAgent.com using port 25.
- Agent has a conversation with the SMTP server. The conversation is an extremely simple set of text commands and responses (see below). Agent tells the SMTP server the address of the sender and the address of the recipient, as well as the body of the parsed message.
- The SMTP server takes the "TO" address (for example, 0126599904@sms.maxis.net.my) and breaks it into two parts:

    1) the recipient name or in this case, users' hand phone number (0126599904)
    2) the domain name (sms.maxis.net.my).

- The SMTP server has a conversation with a Domain Name Server and says, "Can you give me the IP address of the SMTP server for sms.maxis.net.my?" The DNS replies with the one or more IP addresses for the SMTP server(s) that Maxis operates.
- The SMTP server at ScoreAgent.com connects with the SMTP server at Maxis using port 25. It has the same simple text conversation that my e-mail client had with the SMTP server for ScoreAgent, and gives the message to the Maxis mail server. The Maxis server recognizes that the domain name for 0126599904 is at sms.maxis.net.my, so it hands the message to Maxis's POP3 server, which puts the message in 0126599904 message box and send it as SMS to user's hand phone.

ADMINISTRATION SECTION

a) Rules Module

This will be a knowledge engineer's task. A rule base will be built containing if-then rules and inferenced by forward chaining reasoning method. The rules' focus is to search the keyword such as desired team's score and the scorer from the HTML text during parsing process

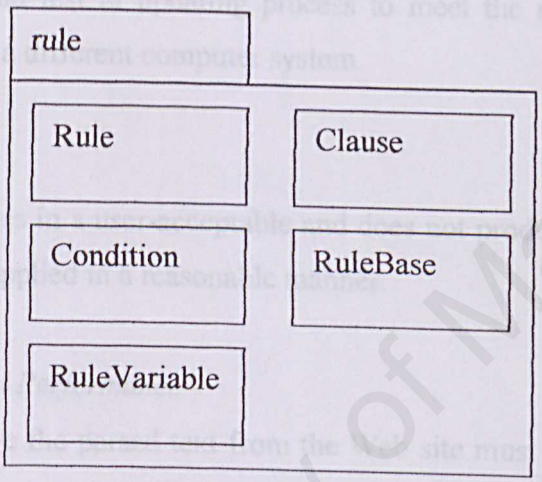


Figure 4-2: The suggested UML package for rule is proposed.

4.1.2 Non-functional Requirement

A non-functional requirement is a description of other features, characteristics and constraints that define a satisfactory system [9]. Below are the non-functional requirements of the system:

i. *Maintainability*

Maintainability is the degree to which the system can be cost-effectively made to perform its functions in a possibly changing operating environment. The system are easy to modify and test in updating process to meet the new request, correcting errors, or move to a different computer system.

ii. *Reliability*

The system operates in a user-acceptable and does not produce dangerous or costly failure when it is applied in a reasonable manner.

iii. *Response Time and Performance*

The time to retrieve the parsed text from the Web site must be within a reasonable time. Then, the formatted text may send to the users in acceptable time. But the performance especially the question "how fast will it reaches user's hand phone?" is very much depending on the telco's SMS service. It will be very slow during daytime but faster at night. Fortunately, soccer matches in England are at night or midnight. Therefore, the response time could consider acceptable.

iv. *User friendliness*

The design of the GUI must able to attract users' focus and easy to use feel. Since the ScoreAgent is focuses on backend tasks, therefore the interface will restricted to user login and register only. The interface for administrator is also provided.

v. *Understandability*

Letting other programmers to understanding the data flow and the coding method.

4.2 THE DATA FLOW DIAGRAM

4.2.1 ScoreAgent DFD – Context Level

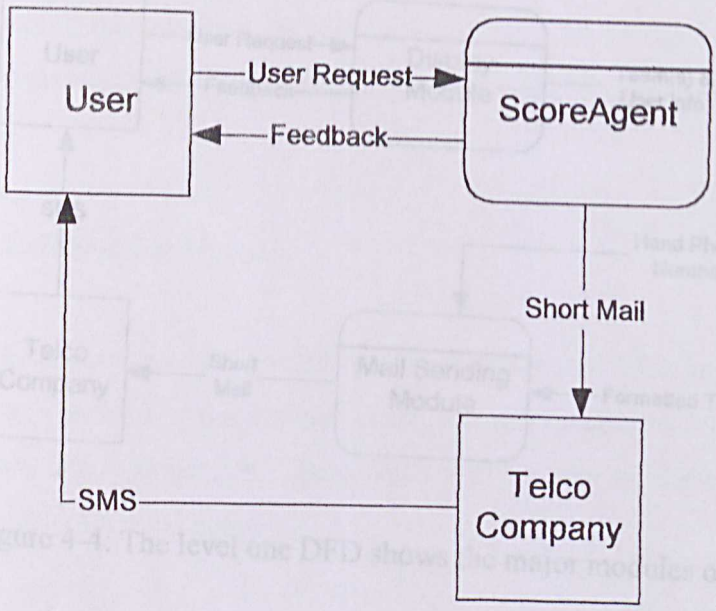


Figure 4-3: Context level of the system.

The context diagram on Figure 4-1 shows all entities that will interact among each other after the system has deployed into the real environment. The data flows are simplified into an easiest way to understand the flow of the system.

From the DFD above, the agent acts as an "agent" for user and the telco company as well. It serves the user where it will direct the user's requests to the telco company.

4.2.2 *ScoreAgent DFD –Level One*

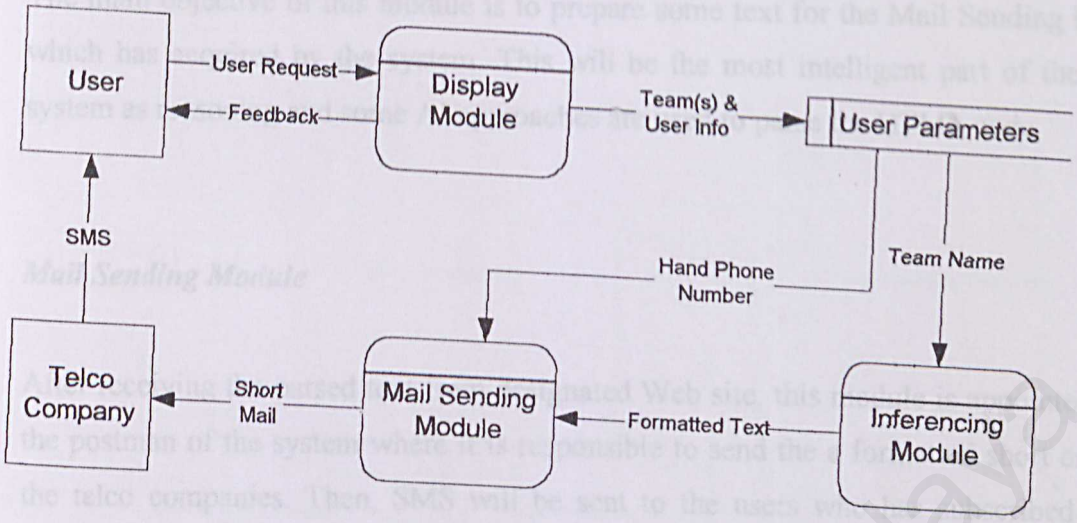


Figure 4-4: The level one DFD shows the major modules of ScoreAgent.

DESCRIPTIONS:

Display Module

This is a user interface that consists of two main screens:

i. *User Input Screen*

To register. It also contain an entry page to pass parameters such as user's hand phone number and preferred team to the database (named User Parameters above) in server.

i. *Output Screen*

A feedback page from administrator to users.

Inference Module

The main objective of this module is to prepare some text for the Mail Sending Module which has acquired by the system. This will be the most intelligent part of the whole system as reasoning and some AI approaches are used to parse the HTML code.

Mail Sending Module

After receiving the parsed text from designated Web site, this module is appointed to be the postman of the system where it is responsible to send the a formatted short email to the telco companies. Then, SMS will be sent to the users who has subscribed to the particular telco company.

4.2.3 *ScoreAgent DFD – Level Two (Inference Module)*

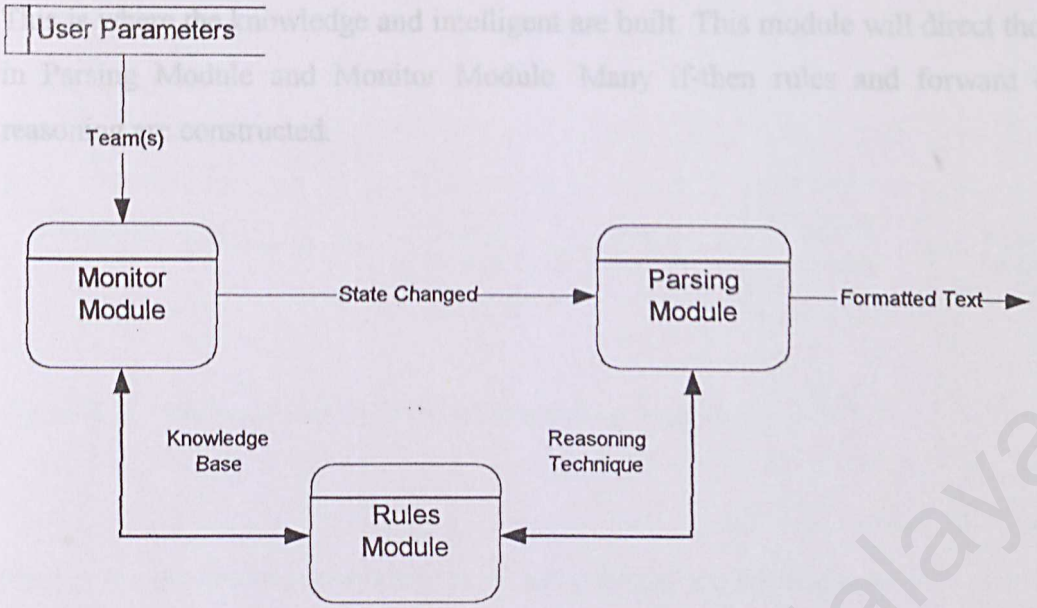


Figure 4-5: The level two DFD for Inference module.

DESCRIPTIONS:

Monitor Module

Keep monitoring the designated Web site for changes. If the content has changed, an event is sent to the Parsing Module to wake it up for subsequent action. It needs the knowledge base from Rules Module during the monitoring process to make itself intelligent enough to determine when to fire the event to other module.

Parsing Module

After it has been waken up by other module, it starts to get its task done by parsing for desired keywords (such as team name, player who scored and so) from the Web site. It will get the knowledge and reasoning technique from Rules Module. After the parsing is done, the text is sent to Mail Sending Module for further processing.

Rules Module

This is where the knowledge and intelligent are built. This module will direct the process in Parsing Module and Monitor Module. Many if-then rules and forward chaining reasoning are constructed.

4.2.4 *ScoreAgent DFD –Level Two (Mail Sending Module)*

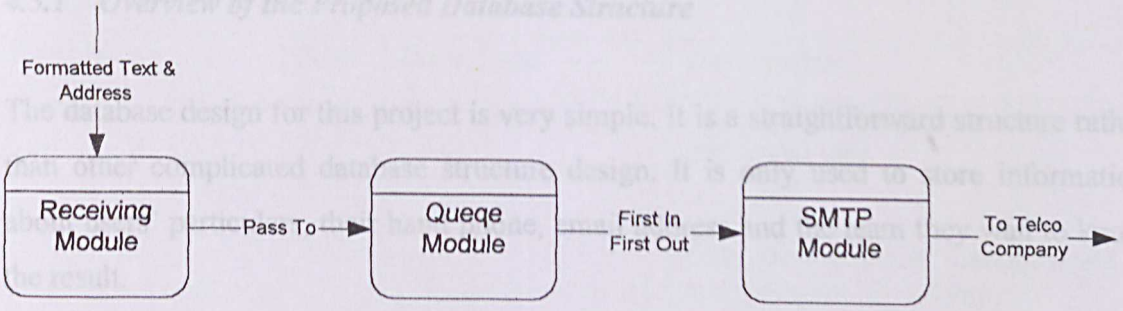


Figure 4-6: The level two DFD for Mail Sending module.

Most of the classes and methods in java.mail package are utilized here:

DESCRIPTIONS:

Receiving Module

Receiving the formatted text together with the address (address to the user's hand phone number). Act as a template.

Queue Module

Implementing First In First Out (FIFO) to manage the sending.

SMTP Module

Sending the text to telco's mail server. The SMS will be sent from telco's mail server which is not in the concern for this proposed project.

4.3 DATABASE DESIGN

4.3.1 Overview of the Proposed Database Structure

The database design for this project is very simple. It is a straightforward structure rather than other complicated database structure design. It is only used to store information about users' particulars, their hand phone, email address and the team they wan to know the result.

Since the database is not used for generating reports or creating entry forms, therefore the structure of the database will be simple but clear about its functionality. It is a 1-to-many relationship between the two tables where sometimes users may want to know more than one team's result. With this structure design, users can receive more than one SMS at a time if the selected matches are held at the same period.

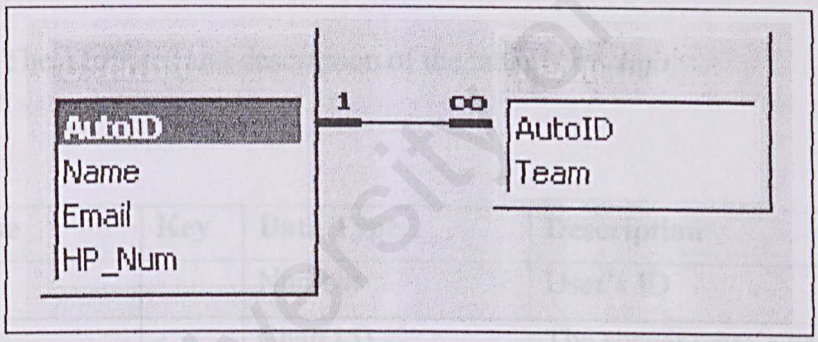


Figure 4-7: The 1 to many relationship.

4.3.2 *ScoreAgent Data Dictionary*

User_Info

Field Name	Key	Data Type	Description
AutoID	Yes	Auto Number	A unique ID number for each of the registered user. Easy identification, searching and accessing during coding.
Name	-	Char(20)	User's name or nickname.
Email	-	Char(20)	User's email address.
HP_Num	-	Char(30)	User's hand phone number with the mail server's name. For example, 0126599904@sms.maxis.net.my It is the address which will be sent to the telco's mail server.

Table 4-8: The attributes and description of the table *User_Info*.

Team

Field Name	Key	Data Type	Description
AutoID	-	Number	User's ID
Team	-	Char(15)	The soccer team's name.

Table 4-9: The attributes and description of the table *Team*.

4.4 SYSTEM USER INTERFACE

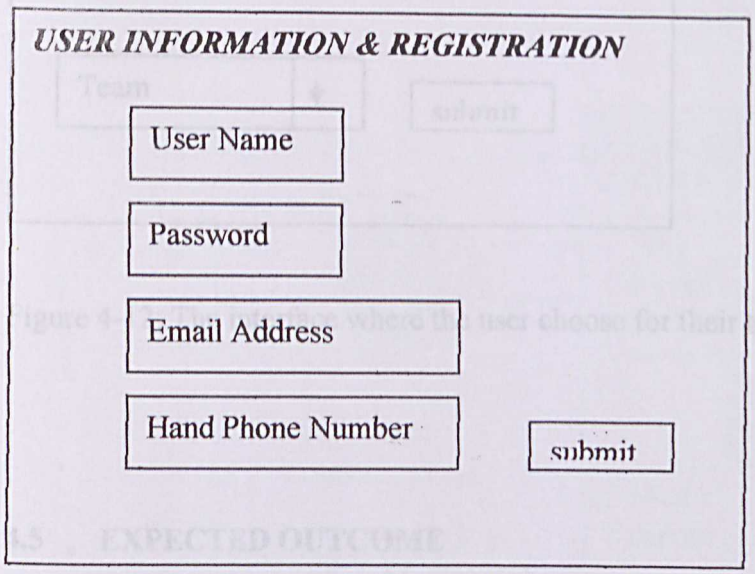


Figure 4-10: The interface where the user needs to register and submit information.

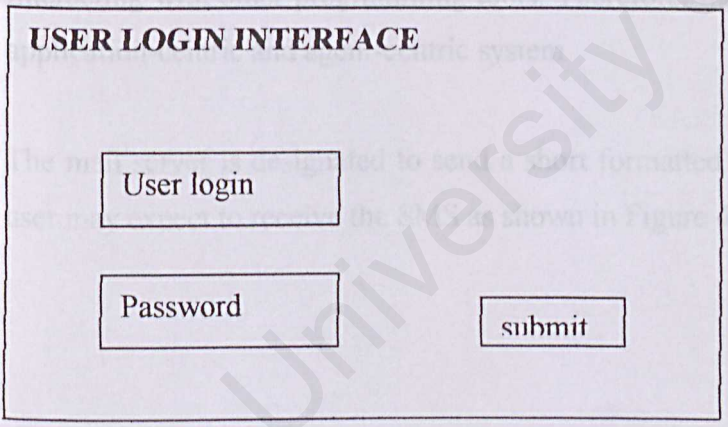


Figure 4-11: The interface where the user try to login.

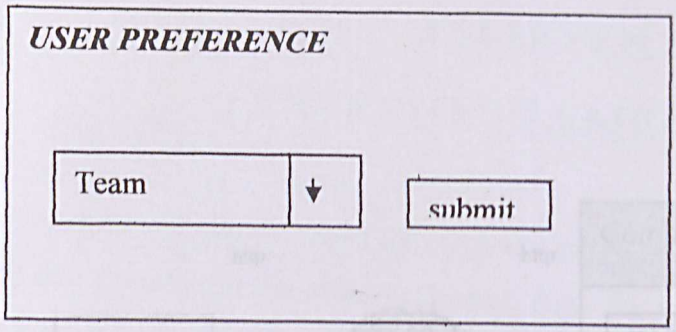


Figure 4-12: The interface where the user choose for their team to be monitored

4.5 EXPECTED OUTCOME

The diagram shown on Figure 4-11 is another way to have an overview look at the proposed system. In general, the proposed system is a Web base system. The cartoon character represents the ScoreAgent. It is clear the ScoreAgent resides in the server and interacting with other programming tools. Therefore, the proposed system is a hybrid of application-centric and agent-centric system.

The mail server is designated to send a short formatted mail to the telco companies and user may expect to receive the SMS as shown in Figure 4-12.

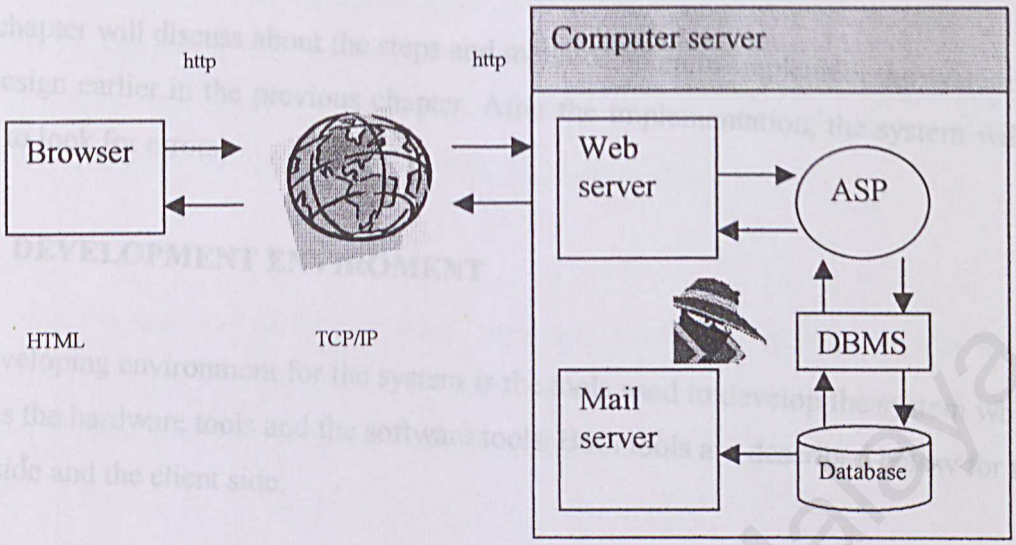


Figure 4-13: The environment where the agent resides.

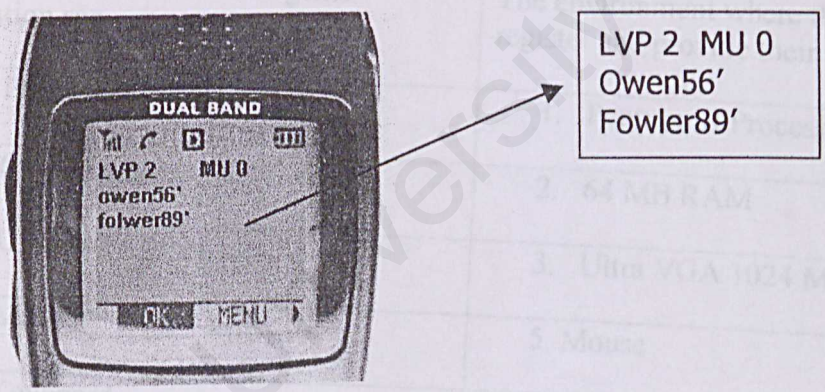


Figure 4-14: Expected SMS received by user's hand phone.

CHAPTER 5
SYSTEM IMPLEMENTATION

This chapter will discuss about the steps and methods taken to implement the system that was design earlier in the previous chapter. After the implementation, the system will be tested to look for errors.

5.1 DEVELOPMENT ENVIROMENT

The developing environment for the system is the tools used to develop the system which ncludes the hardware tools and the software tools. Both tools are described below for the erver side and the client side.

1.1 Hardware Tools

erver	**Client**
escription: ie environment where the agents plication run.	Description: The environment where the users login to register and provide their information.
1. Pentium II Processor.	1. Pentium II Processor.
2. 64 MB RAM	2. 64 MB RAM
3. Ultra VGA 1024 Monitor	3. Ultra VGA 1024 Monitor
4. Mouse	5. Mouse
. Keyboard	6. Keyboard
Local Area Network	7. Modem

5-1: Hardware Requirements Tools.

5.1.2 *Software Tools*

Windows 2000 OS

The platform used to run the application. It also has an extra function compare to the predecessor where it can act as a web server which is the Internet Information Server, IIS to keep created web pages. This function is important to able the application to connect to the mock web sites during presentation and during debugging.

JBuilder 5

It provides the easy way to develop a Java application with its powerful and user friendly environment. A lot of time has been saved during the interface development because JBuilder 5 has a template called designer where the developers can create a rapid interface design.

Java Package

Javamail™ Package is downloaded from the official java site and has imported into the system. This will enable the system to send mail via a declared SMTP port.

ASP and MacroMedia Dreamweaver

It is used to process the web pages upon users' request which require users to input their particulars and store the information into the database.

5.2 SYSTEM IMPLEMENTATION

Implementation comprises of the system design structure to a computer readable system. The system will be evolved from scratch design to a run able application. There are several implementations for this system.

5.3 INTERFACE AND DATABASE IMPLEMENTATION

The web site is created in order to allow user keys in their personal information as shown in Figure 5-3 and change team preference to know the desired team's live score as shown in Figure 5-4. The pages are developed using Macromedia Dreamweaver Ultradev and the database is implemented with Microsoft Access 2000 as the DBMS. Users' information is stored in the database named UserInfo_DB.

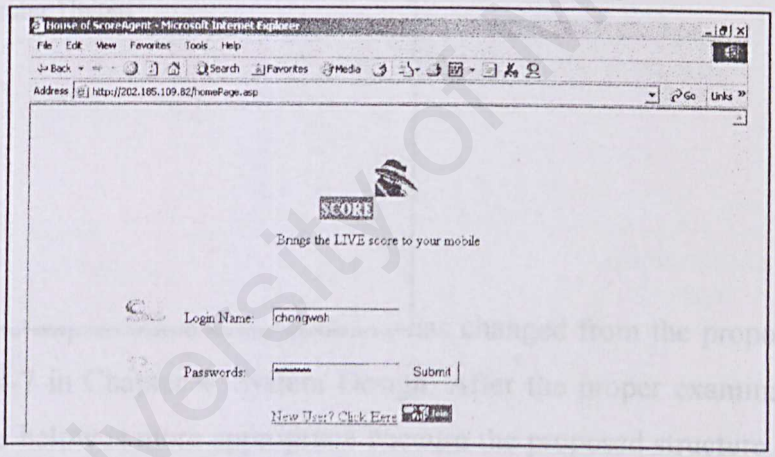


Figure 5-2: The ScoreAgent homepage.

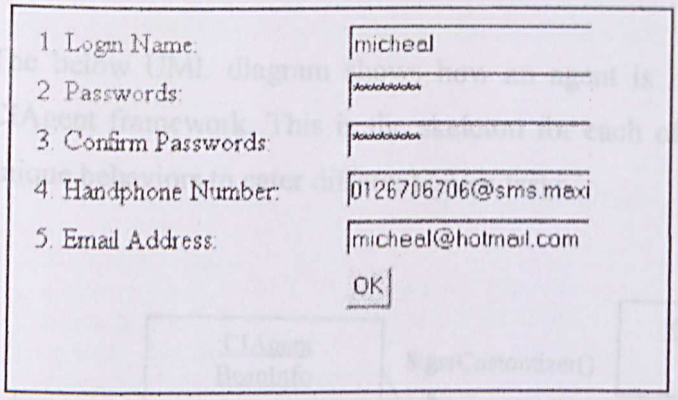
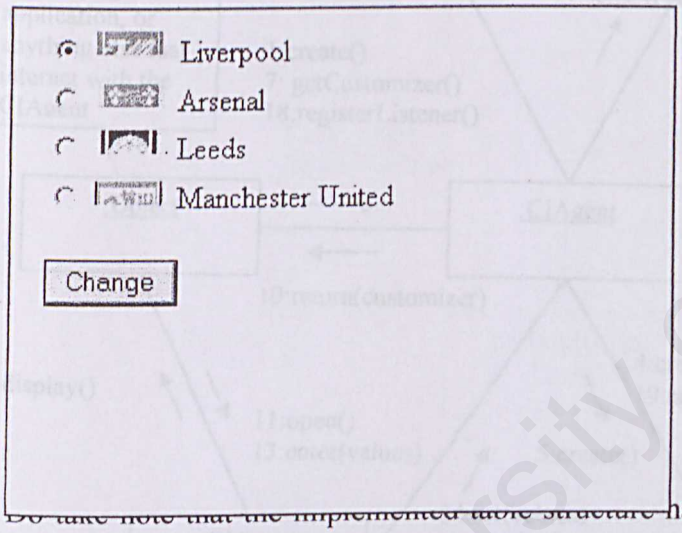


Figure 5-3: Users' personal information.



Do take note that the implemented table structure has changed from the proposed table structure at Figure 4-7 in Chapter 4: System Design. After the proper examination, the single table structure below is more appropriate because the proposed structure is having redundancy problem. This is a sufficient design to cater any data insertion from users for this application.

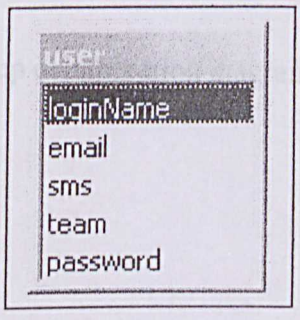


Figure 5-5: The implemented single table named "user".

5.4 INTELLIGENT AGENT FRAMEWORK IMPLEMENTATION

The below UML diagram shows how an agent is initialized or born. It is called the CIAgent framework. This is the skeleton for each of the agents except with their own unique behaviors to cater different given tasks.

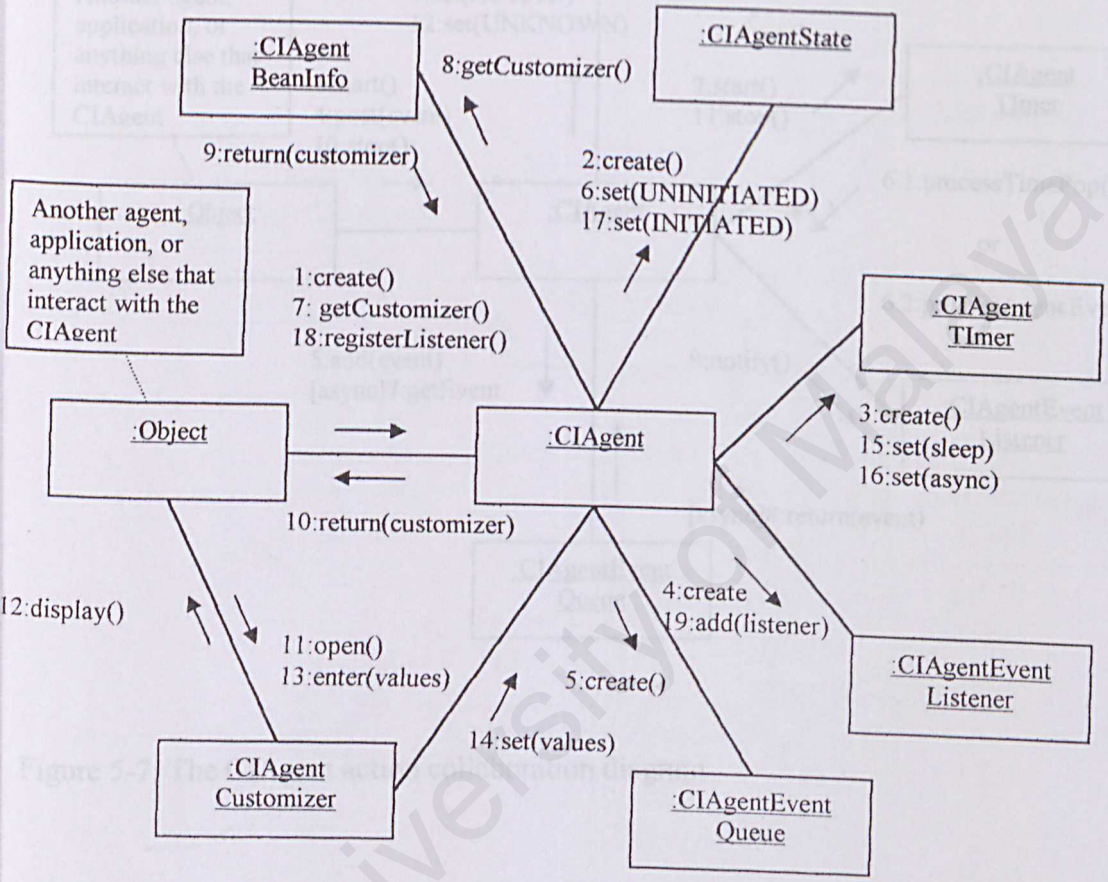


Figure 5-6: The CIAgent start-up collaboration diagram.

The below UML diagram shows the lifecycle of a CIAgent framework where the agents have inherited. The diagram shows the common behaviors on each agent.

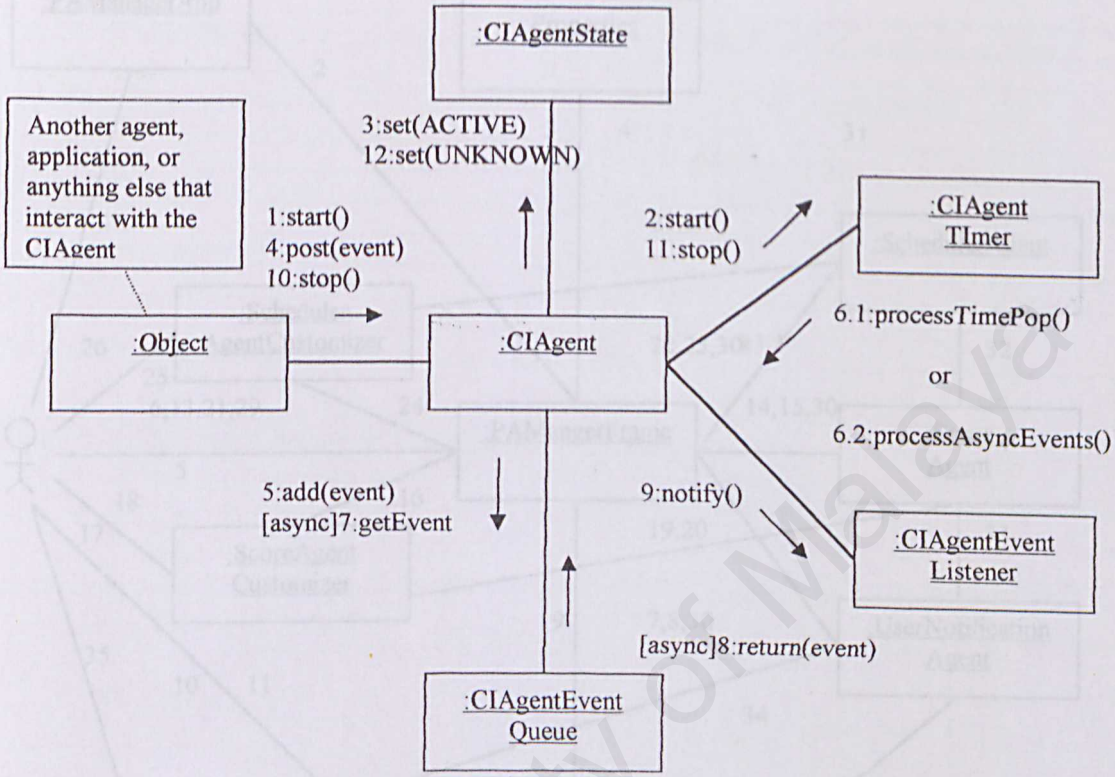
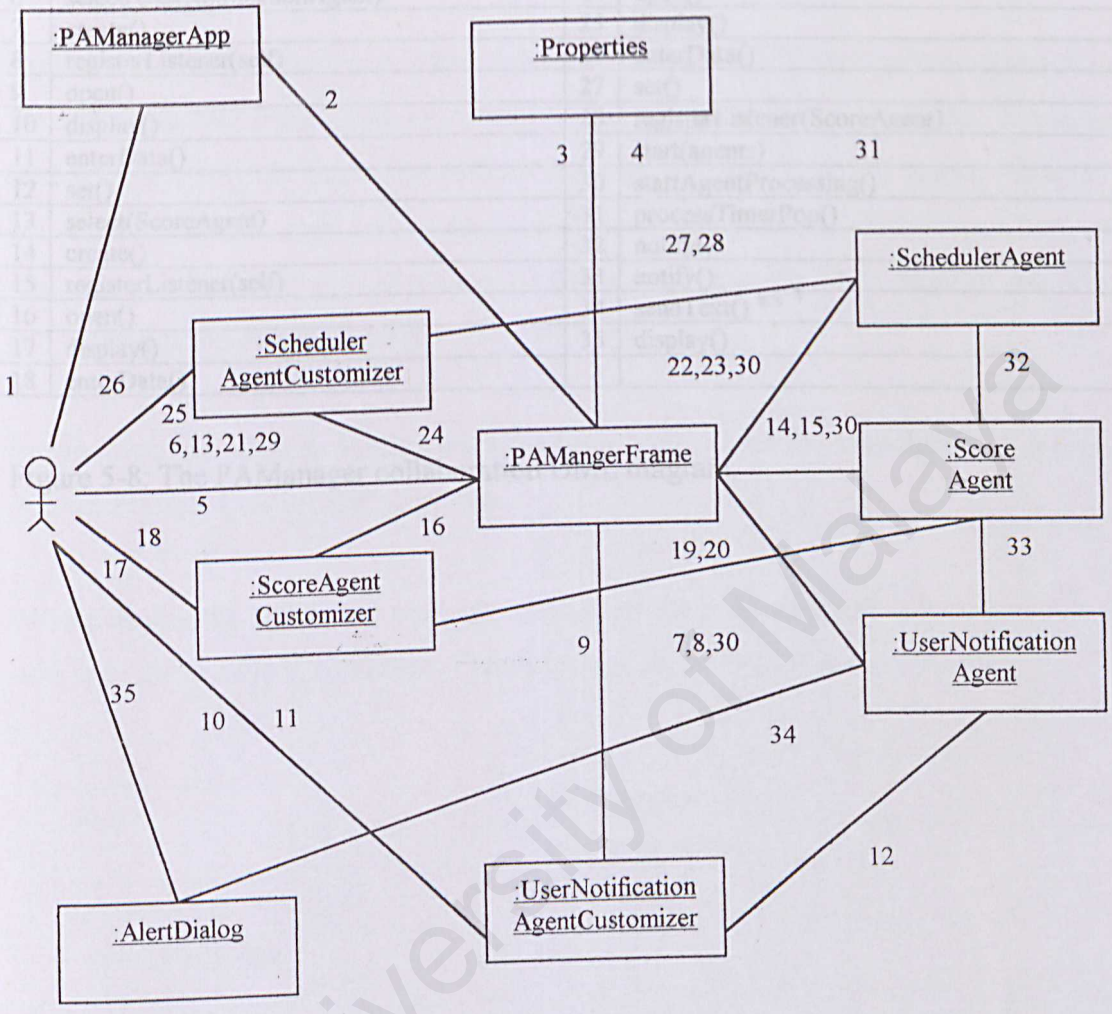


Figure 5-7: The CIAgent action collaboration diagram

The UML diagram shows how all the agents (ScoreAgent, SchoduleAgent and UserNotification Agent) interact with each other by passing events.



1	start()	19	set()
2	create()	20	registerListener(UserNotificationAgent)
3	readProperties()	21	select(SchedulerAgent)
4	return(availableAgents)	22	create()
5	display()	23	registerListener(self)
6	select(UserNotificationAgent)	24	open()
7	create()	25	display()
8	registerListener(self)	26	enterData()
9	open()	27	set()
10	display()	28	registerListener(ScoreAgent)
11	enterData()	29	start(agents)
12	set()	30	startAgentProcessing()
13	select(ScoreAgent)	31	processTimerPop()
14	create()	32	notify()
15	registerListener(self)	33	notify()
16	open()	34	sendText()
17	display()	35	display()
18	enterData()		

Figure 5-8: The PAManager collaboration UML diagram.

5.5 RULE BASE REASONING IMPLEMENTATION

5.5.1 Creating Rules in the Rule Base

To represent rules as the knowledge in the rule base using object-oriented approach does making sense. We should treat everything which related to rule as an object. The general terms in rule are rule name, condition (i.e.: equals, more than...), clause, antecedent, consequent, variable, right hand side and left hand side. We treat all the terms here as an object which has their own attributes, methods and behavior. Thus, each of the objects can behave as we wanted them to be.

In this case, we want to build a set of rules which act as the knowledge to recognize English Premier League, EPL matches.

After defining the terms, the next step is to bind them up as a unit which is a rule and store it in the knowledge database. The following shows how it does.

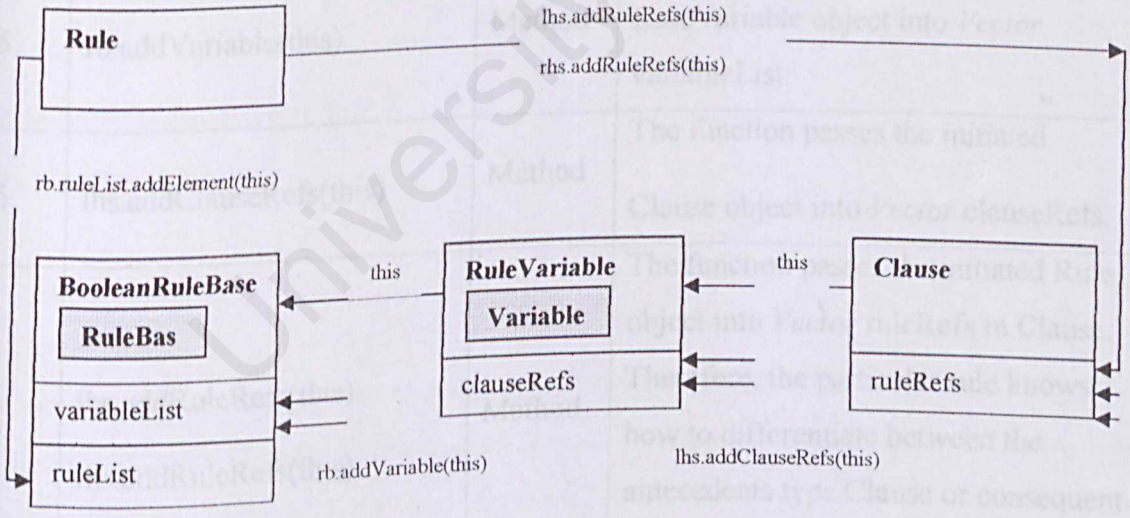


Figure 5-9: The diagram shows the definition of a rule in the rule base.

Some important terms to interpret the graphical representation above:

a) The rectangles represent classes.
b) The darken rectangles are the *superclass* for some classes.
c) The arrows show where the corresponding classes refer to.
d) The bolded words are the class name.
e) The words in the boxes are the *Vector*, a data structure in Java which store objects. It is a dynamic array structure.

No.		Type	Description
1	variableList	Vector	Contains all the initiated variables in the rule base.
2	ruleList	Vector	Contains all the initiated rules in the rule base.
3	clauseRefs	Vector	Contains all the initiated clauses which the rule variables refer.
4	ruleRefs	Vector	Contains all the initiated rules which the clause refer to.
5	rb.addVariable(this)	Method	The function passes the initiated RuleVariable object into *Vector* variableList.
6	lhs.addClauseRefs(this)	Method	The function passes the initiated Clause object into *Vector* clauseRefs.
7	lhs.addRuleRefs(this) rhs.addRuleRefs(this)	Method	The function passes the initiated Rule object into *Vector* ruleRefs in Clause. Therefore, the particular rule knows how to differentiate between the antecedents type Clause or consequent type Clause.
8	rb.ruleList.addElement(this)	Method	Contain all the rules objects.

Table 5-10: Important methods and data members description.

5.5.2 *Defining a Rule Base*

First, we need to instantiate a new rule base with a name in the *BooleanRuleBase* class. In this case, the rule base will be named as "*EPL Team Information*". The "EPL Team Information" rule base will contain all the knowledge about the team name in the English Premier League, EPL. The examples of rules are shown below with the intention to create the rule named *Lvp1* and *Ars1*.

Rule Base Name: "EPL Team Information"

Lvp1 : *IF teamName = "Liverpool"*
AND oppTeam = "Chelsea"
THEN ePL = " true"

Ars1 : *IF teamName = "Arsenal"*
AND oppTeam = "ManU"
THEN ePL = " true"

Figure 5-11: Rules example.

Defining Variables

Then, instantiate correspondent variables *teamName*, *oppTeam* and *ePL* as RuleVariable objects and refer to each of the rule as shown in Figure 5-9 and Figure 5-11. In other words, that particular rule and the rule base "*EPL Team Information*" knows what its valid variables are and will store them in the Vector *variableList*.

Defining Clauses

Each variable (i.e.: *teamName, oppTeam*) needs to know which clause is bounded to. With that, the variable knows its valid string value (i.e.: "Liverpool") and the condition (i.e.: =). Therefore, after a new Clause object is instantiated, it will refer back to its variable. From Figure 5-11, the clauses are:

teamName ="Liverpool"	;	*teamName = "Arsenal"*
oppTeam = "Chelsea"	;	*oppTeam = "ManU"*
ePL =" true"	;	*ePL =" true"*

Figure 5-12: Example of clauses which will refer back to the correspondent created variable object.

Defining a Rule

The previous instantiated clauses will in turns refer to their correspondent rule. As an example the clause object *teamName ="Liverpool", oppTeam = "Chelsea" and ePL =" true"* will be bound to the rule *Lvp1* after the clauses pass itself to the corresponding rule.

After a rule has been instantiated, it will pass itself to the Vector *ruleList* in *BooleanRuleBase* class. This process will continue as the knowledge engineer creates a desired number of rules as the knowledge database to keep in *Vector ruleList*.

5.6 FORWARD CHAINING INFERENCE IMPLEMENTATION

After all the needed rules were defined, it is considered that we are having a set of knowledge about the EPL match. The rules needed to be inferred by the inference engine using forward chaining method before it can determine or validate an EPL match.

When the ScoreAgent asks the forward chaining inference engine for validation, the algorithm goes as follow:

a) *Load the rule base into the inference engine and load any facts from the knowledge base into the working memory.*

The rule base is loaded after it has been defined as discussed. The instantiation of a rule base starts as shown below. *rb* is the new instantiated object from *BooleanRuleBase* class.

```
BooleanRuleBase rb = new BooleanRuleBase("EPL Teams Information");
```

To initiate the *RuleVariable*,

```
RuleVariable teamName = new RuleVariable(rb, "teamName");
```

b) *Add any additional initial data into the working memory.*

To load facts from the knowledge Add any additional initial database into the working memory. The parsed string name *strLvp* is the string for home team.

```
teamName.setValue (strLvp);
oppTeam.setValue (strLvpOpp);
```

The *setValue (strLvp)* method will set the current variable *teamName*'s and *oppTeam*'s value to string *strLvp and strLvpOpp* respectively.

c) *Match (a function) the rules against the data in the working memory and determine which rules are triggered, meaning that all of their antecedent clauses are true. This set of triggered rules is called the conflict set.*

The inference process starts with the statement,

```
rb.forwardChain ();
```

All the available rules in the rule base are *check (a function)* for the truthness of its antecedents *Clause*. For the rules which all its antecedents Clause is tested true, they will be returned and stored in a *Vector* type data structure named *ConflictRuleSet*. The rules in *Vector ConflictRuleSet* are the potential candidates to be fired.

d) *Use the conflict resolution procedure to select a single rule from the conflict set.*

From the *Vector ConflictRuleSet*, a conflict resolution procedure is used to *select (a function)* the best possible rule from a set of potential candidate rules. The approach is to select the rule which has the highest number of antecedents.

e) *Fire(a function) the selected rule by evaluating the consequent clause(s); either updates the working memory if it is a fact-generating rule, or calls the effector procedure, if it is an action rule.*

The best rule has the maximum number of antecedents. Now, its consequent Clause's string value is determined as the data or information for user. The determined string value is the result for the inference process provided there is no other rule in the *Vector ConflictRuleSet*. For this system, the inference process will only reach at this stage because all the constructed rule are simple.

f) *Repeat steps 3, 4, and 5 until the conflict set is empty.*

To determine the outcome for the inference process.

5.7 WEB PAGE PARSING AND MONITORING IMPLEMENTATION

Tasks of ScoreAgent:

a) Connect to the Web Pages

It is an easy task to connect to the desired web pages on the internet by using the java.net package. The code is shown as below.

```
URL ttLvp = new URL("http://202.185.109.82/_liverpool.html");
BufferedReader in = new BufferedReader(new InputStreamReader(ttLvp.openStream()));
```

* Notes: The URL address shown http://202.185.109.82/_liverpool.html is the mock web page that has been created for the testing and presentation purpose.

b) Parsing the Connected HTML codes

Refer to the *BufferedReader* object above, method *in.readLine()* is used to parse the HTML code line by line until the end of the connected page. Therefore, we can search for the specific strings such as team name and the current score from the connected page using a while loop.

After the parsing, the ScoreAgent will check for score changes by comparing the current parsed score strings with the previous score strings. If there is a different, the current parsed strings will be stored as the "previous score strings" for the next parsing.

Then, ScoreAgent passes the parsed team string and opponent string from the web page to the inference engine to verify whether it is an EPL soccer match. If a rule's antecedents match all the passed strings, the rule's consequent Clause is set to "true" else null or "false".

c) Query the Database

ScoreAgent will query for the user's team preference from the connected database. Then, the list of users for the corresponding team preference is returned to the Mail Sending Module.

d) Sending Mail

In this module, the system will send an email or SMS to user according to the returned list together with the latest score and period. Details on the following section.

5.8 MAIL SENDING IMPLEMENTATION

Javamail™ Package is downloaded from the official java site to enable the system to send mail. Some configuration is needed to set the path at the system to a required *Java Archive* File, *JAR* file. The *JAR* files act as the library to the system which enable the Java Compiler to recognize the methods in Javamail™ Package during the sending session take place.

The Javamail™ Package allows you to:

a) Set the subject or title for the message
b) Attach the message that we wanted to send
c) Send message to single or multiple recipients
d) Declare the sender's mail address

A SMTP port must be declared before you can send a mail to the outside world. At the current occasion, the system is using faculty's SMTP address. We can download a mail server freeware from the internet as our own mail server by initiating the domain as the IP address of the computer. Either way will do.

CHAPTER 6

SYSTEM TESTING

6.1 INTRODUCTION

Software testing is one of the main phases in the Waterfall Life Cycle model. In this phase, the process of testing and debugging are done to detect defects and bugs of a system. These processes are usually done incrementally with system development.

This phase is also often referred to as Verification and Validation (V & V). Verification refers to the set of activities that ensure the software correctly implements a specific function. Validation refers to a different set of activities that ensure the software has been built is traceable to user requirements. A successful test is one in which no errors are found.

The objectives to test this system are:

a) To reveal inference error by the inference engine based on the rule base.

b) To compare the expected outcome with the actual outcome. Eventually, debug it to enhance it functionality and capability.

c) To ensure the SMS or email is sent by the agent and has been received by users and according to the users' preference in the database.

6.2 UNIT TESTING

Unit testing focuses verification effort on the smallest unit of software design that is the software component or module.

6.2.1 Testing Display Module and Database

This is referring to the user login and database testing to ensure that every data which user insert from the web page is stored accurately and correctly to the correspondence database. A set of sample raw data is created for the testing purpose in this module. The process includes iterate the checking on the duplicated data in database to ensure that every entered data is valid and ease the redundancy and duplication problem. The control objects such as radio button, combo box and text field are tested too to ensure the correct functionality respectively.

6.2.2 Testing Mail Sending Module

A set of user's email address and hand phone data is prepared for ScoreAgent during retrieving the user information from database. Then, the email accounts are all checked to verify that the mock users have received the message from ScoreAgent.

loginName	password	email	sms	team
chongwah	chongwah	chongwah@hotı	0126599904@s	liverpool
suyee	suyee	suyee79@hotm	0127272916@s	liverpoolx
hoe	hoe	chongwah@hotı	0126558317@s	manx
kok	kok	cekcool@hotma	0126706176@s	liverpool
whoong	whoong	wongpak@hotm	0123043147@s	arsenalx
wongpak	wongpak	wongpak@hotm	0126710760@s	arsenal
tiangseng	tiangseng	seng_1@hotma	0123512691@s	leedsx
chungyl	chungyl	chungyl@hotma	0122619119@s	man
pangjs	pangjs	pangjs@yahoo.	0123638338@s	leeds

Figure 6-1: The testing set of data from database.

The redden rows are the errors created. This is to ensure these users won't receive emails or SMS from the agent. Other users should get some feedback from the system.

6.2.3 *Testing Parsing and Monitoring Modules*

In this module, a study of a certain web pages' HTML codes is necessary. This is to identify the specific strings on a specific section that we wanted to parse. A serial of testing has been done to parse the wanted string from the specific web page.

```
<html>
<head><title>Liverpool Score</title></head>
<body>

<tr border=1 >
<TD align=right>Liverpool</TD>
<TD align=right>0</TD>
<TD align=left>0</TD>
<TD align=left>Chelsea</TD>
<TD align=left>latest match</TD>
</tr>

</body>
</html>
```

Figure 6-2: The source of the "mock" HTML page.

The bolded strings are the strings that the agent should retrieve. Therefore, to study the structure of the HTML codes for some particular web pages is essential in order to get the right results, which are the strings we want.

6.2.4 *Testing Rules Module*

Times of testing have been done on the forward chaining algorithm to ensure that it could produce new data and eventually the desired result. This includes preparing a set of rule to be inferred.

```
Lvp1  :  IF teamName ="Liverpool"
         AND oppTeam = "Chelsea"
         THEN ePL =" true"
```

```
Ars1  :  IF teamName = "Arsenal"
         AND oppTeam = "ManU"
         THEN ePL =" true"
```

Figure 6-3: Samples of rule from rule base.

To test whether the inference process is working or not, we must make sure that each of the rules is a piece of knowledge in the rule base. When the rule, say *Lvp1* is being deleted, we try to pass the corresponding parsed strings (*Liverpool* and *Chelsea*) from the web to the inference engine. If the inference engine unable to determine it is an EPL match, then the inference process if correct. Else, there's a bug. Hereby, we can say that the level of knowledge in rule base has been decrease. Therefore, adding more rule will enrich the agent's knowledge and vice versa.

6.3 INTEGRATION TESTING

For this project, a bottom-up approach has been used. Bottom-up integration testing begins construction and testing with modules at the lowest levels of the system and then moving upward to the modules at the higher levels of the system.

Since all the modules have been tested and have been declared bug free, the modules will combine one by one according to the called modules. Each integration is checked again to ensure there is no error.

6.5 SYSTEM TESTING

System testing is a series of different tests designed to fully exercise the software system to uncover its limitations and measure its capabilities. The objective is to test an integrated system and verify that it meets specified requirements.

There are several types of system testing that are worthwhile for a software system.

a) *Rule Testing*

It is a test during the run time environment where the complete set of rules is loaded. If the inference engine can differentiate the English Premier League matches and non- English Premier League matches, then it has considered success.

b) *Performance Testing*

The purpose of this testing is to test the run-time performance of software within the context of an integrated system. This will show the actual outcome which is the message received and compare to the expected outcome.

c) *Web site Testing*

This is a test on the user login web site. The web application should be able to store information from users accurately into the database. The flows and links between web pages should be logic to user. Else, they might lost amidst the surfing.

6.6 TESTING ANALYSIS

Overall, the system runs smooth. The rule base is able to recognize the EPL match. The messages could reach users' mobile correctly with the match latest result. as a conclusion, all the objectives have been achieved.

CHAPTER 7
SYSTEM EVALUATION & CONCLUSION

7.1 SYSTEM EVALUATION

During the period of coding and implementation of this system, various problems were encountered. These problems were solved through research and studies in fields such as the Internet, Bulletin Board Website, journals and reference book. The system's strengths, limitations, and future enhancement were identified.

7.2 PROBLEMS ENCOUNTERED AND SOLUTIONS

Problems are everywhere and so does in every system. Several problems encountered throughout the development of this system. These include:

a) Difficulty in Choosing Development Technology and Tools

There are many software tools available to develop an Intelligent Agent system. Choosing a suitable technology and tools was a critical process as all tools possesses their own strengths and weaknesses. In addition, the availability of the required tools for development was also a major consideration. Eventually, Java has been chosen of its strength in internet connectivity and cross platform behavior. From the internet, there are thousands of useful packages which we might need. Therefore, we only have to learn about the package's methods to program rather than create all the codes by ourselves. It saves us a lot of time.

b) *Lack of Knowledge in the Concept of Constructing an Intelligent Agent*

Theoretical has never been hard. All the problems and doubts will emerge once the development and design starts. It's hard to find a good Intelligent Agent framework from the internet or book. Without a good framework or skeleton, the structure of the Intelligent Agent and the rest of the process are hard to proceed. This system is using a framework which had recommended by many web sites.

c) *Lack of Knowledge to build a Rule Base with Forward Chaining Inference Engine Using Java*

Though I have been taught about what is rule base and forward chaining in the class, but to construct it using Java is a new thing for me. The design on how the rules should be bound to each other, how a rule can recognize its own rule variables and the values respectively take the most of the time during the development of this system. Internet provides a rich source about the algorithm and the correct way to construct it using Java.

d) *Lack of Knowledge in ASP*

Previous knowledge in traditional, two-tier non web-based system does not seem to solve the lack in web-based system. Since there was no prior knowledge of programming in ASP and VBScript, there was an uncertainty on how to organize the codes in a web page. These new programming languages and concepts were never taught before and to implement such as application requires a fair grasp of the languages. These programming approaches seem to be totally different from the traditional programming languages.

e) HTML Parsing Problem

There area a lot of sports web sites that provide the live score to the surfers who want to know the latest EPL match result. Different web site has their own way to present the score with the different HTML coding. Therefore, study about the complexity of the HTML codes is essential in building statement using Java to parse the desired string. Eventually, I had decided to choose www.teamtalk.com as my ScoreAgent's monitoring page because the web site's layout format is not frequently changed. This makes the string search from the HTML becomes more predictable.

f) System Testing Problem

As we know, the EPL matches are not held everyday! But I need to test my system very frequently. Therefore, I had to create a mock web site as my testing template in the Windows 2000 Internet Information Server, IIS to run the test locally on the PC before I can move the completed system to monitor the real web site on the internet.

7.3 SYSTEM STRENGTHS

During the development of this project, several system strengths were identified and described as follow:

a) *User Friendliness and Easy to Use Interface*

Some useful Graphical User Interface (GUI) such as command buttons, check boxes and drop-down list boxes are created on the web pages and the agent application which could attract the users to navigate through the system and give faster access. This user-friendly interface can minimize learning time for the administrator.

b) *Provide Database Access*

All the data are organized and stored in the form of database using Microsoft Access 2000. It is real time database information and any changes made to the records can be updated instantly by users via internet access.

c) *Autonomous*

After the administrator activated the agents to monitoring the web site, the administrator can leave the rest of the task to the agents. The ScheduleAgent will ask the ScoreAgent to do its task for a determined interval of time. If there is a problem on the system, UserNotificationAgent will pop out a message to administrator about the problem. All the agents are independent. Each of the agents has its own area of task. When a task is not in an agent's concern, the task will pass the event to other agent which is more appropriate for the task.

d) Forward Chaining Inference Engine and Rule Base

The selected reasoning approach enables the searching for the new data from the rule base. Every piece of constructed rule acts as the part of the knowledge and the rules can be edited, added or deleted anytime. As an example, if some of the current teams have been relegated to the First Division or teams from First Division have been promoted to play in EPL, the rules always can be updated without affecting the coding on other modules.

e) Utilizing the SMS service

We know how convenient SMS it is. Therefore, to send the latest score to users' hand phone is a good approach in keeping the users always informed whenever it is, wherever it is. Furthermore, it is totally a free service from this application.

7.4 SYSTEM LIMITATIONS

Owing to the time constraint and the programming language itself, there were some limitations in this system. These include:

a) *Unable to Detect Changes On HTML code*

If the web page that we are referring changes its contents and layout very frequently, the string format in HTML will be different from the previously codes. This will cost a severe problem to the ScoreAgent as it has not been equipped to recognize changes towards the HTML codes. If this does happen, the ScoreAgent will fail in searching for the desired string because HTML describe only how the content should be presented but not the meaning of the page content. Hopefully, the emergence of the eXtensible Markup Language (XML) will solve this problem as it provides its tag means during communication session with other web pages.

b) *SMS Delivery Delay*

There is a delay on the message that we have sent from agent to user. Though this is not ScoreAgent's fault but it would be nice for users to get the message real time. The server on the telco company plays a big part in this matter. Fortunately, their server is off-peak at night and EPL soccer matches always held at night in Malaysia time. Therefore, the delay time is not a very big concern.

7.5 FUTURE ENHANCEMENT

Further development and many new ideas have come about while the system was being implemented. Owing to time constraint and other factors, not all of the ideas could be incorporated into the system. It is hoped that the following aspects could be considered in future:

a) Implement Natural Language Processing into the ScoreAgent

Since the ScoreAgent parses only the content of the HTML codes but not the meaning of the content, it is possible to implement NLP method into ScoreAgent. Thus, it is able to parse for the meaning of the content during the search for the desired strings even the format of the web page has changed.

b) Update the Match Fixture From the Web Automatically

In order to make the agent more autonomous, it may check on the match fixture section on the HTML codes and parse for all the match fixture information which include the day and time. Therefore, the agent can store and organize the information into database or file. The ScheduleAgent may keep track on the time for each parsed match and set the time to inform ScoreAgent about the next start monitoring session (when there is a match).

USER MANUAL

SCORE AGENT README

1.0 ABOUT SCORE AGENT

ScoreAgent will monitor a web site which provides the latest English Premier League matches score. It will keep users inform about the latest score by inference the parsed web pages' content using rule base and by forward chaining. Eventually, the agent will send a mail (result of current score) to users' mobile.

2.0 SETTING UP THE SYSTEM BY ADMINISTRATOR

2.1 System Requirement

Hardware:

1. Intel Pentium Processor II or Compatible and higher
2. 32 MB RAM or higher
3. Internet connection
4. Hard Disk Space of At Least 5 MB of free space

Software:

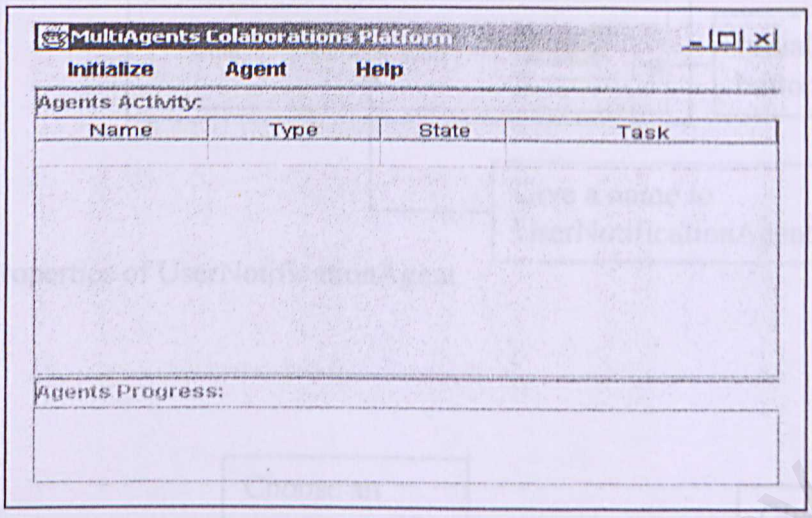
1. Windows 2000 Professional with IIS
2. JDK1.2 (Java Virtual Machine) or higher (downloaded from http://java.sun.com)
3. JavaMail™ package from http://java.sun.com.
4. JDBC – ODBC Connection.

Steps:

1. Install JDK
2. Install JBuilder or other Java IDE application
3. Download JavaMail™ package.
4. Setting path and classpath for the JavaMail™
   - modify the CLASSPATH variable to include *c:\jaf-1.0.1\activation.jar; c:\javamail-1.1.3\mail.jar;*
   - modify the PATH variable to include c:*\jdk1.3\bin* directory;
5. Run the ScoreAgent.

**Note: The following section shows show how to run the system in details.*

1. After the application start, the main interface of the application is shown as follow:



2. To start the agents, follow the steps as the shown in the ***correct sequence and the correct parameter in the customizer boxes***. Click the button Initialize to initialize each agent. Follow the setting as shown especially for the dialog box number 3.

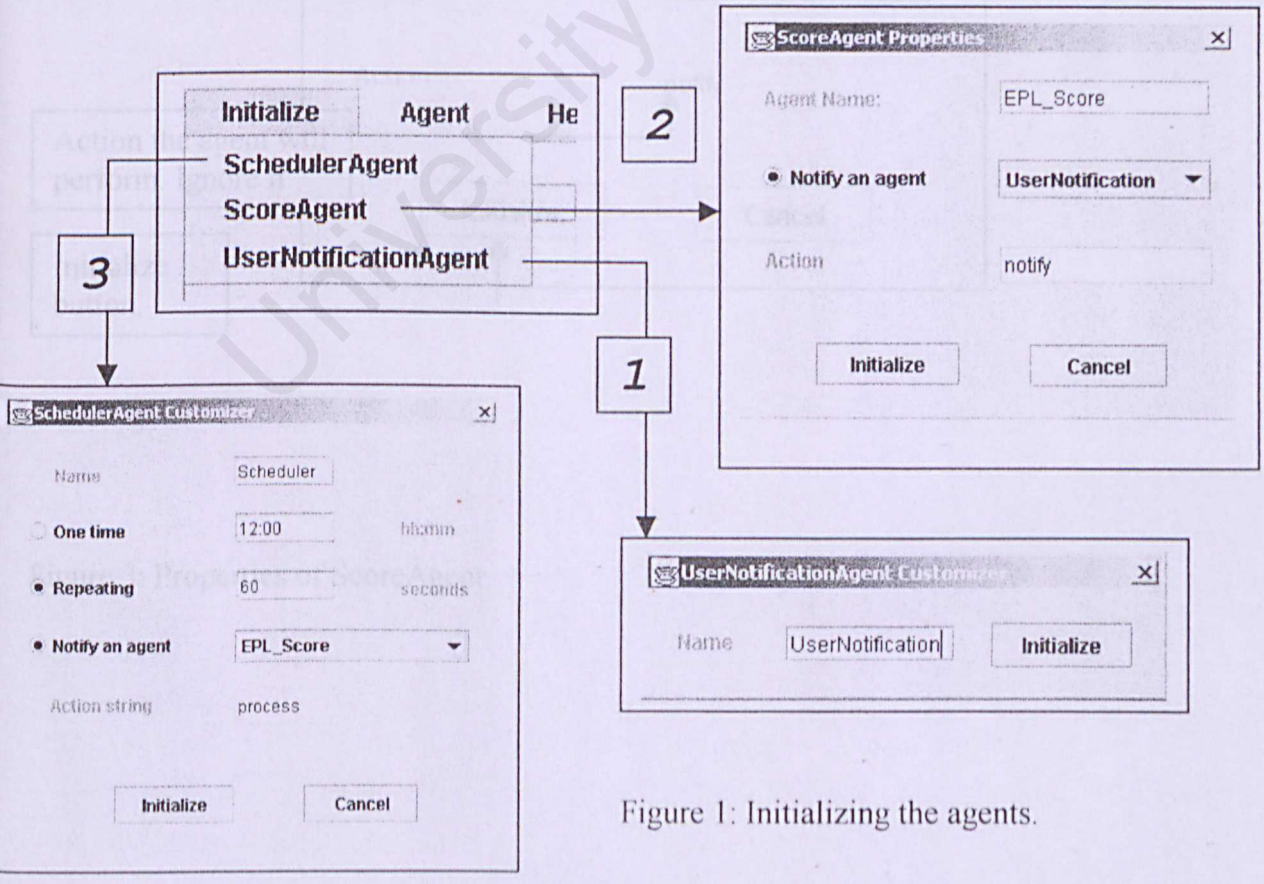


Figure 1: Initializing the agents.

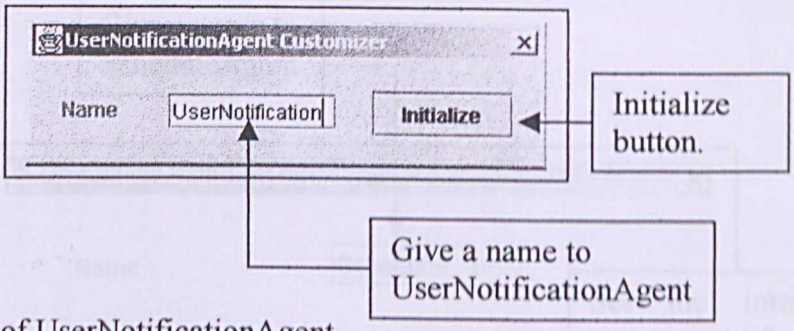


Figure 2: Properties of UserNotificationAgent.

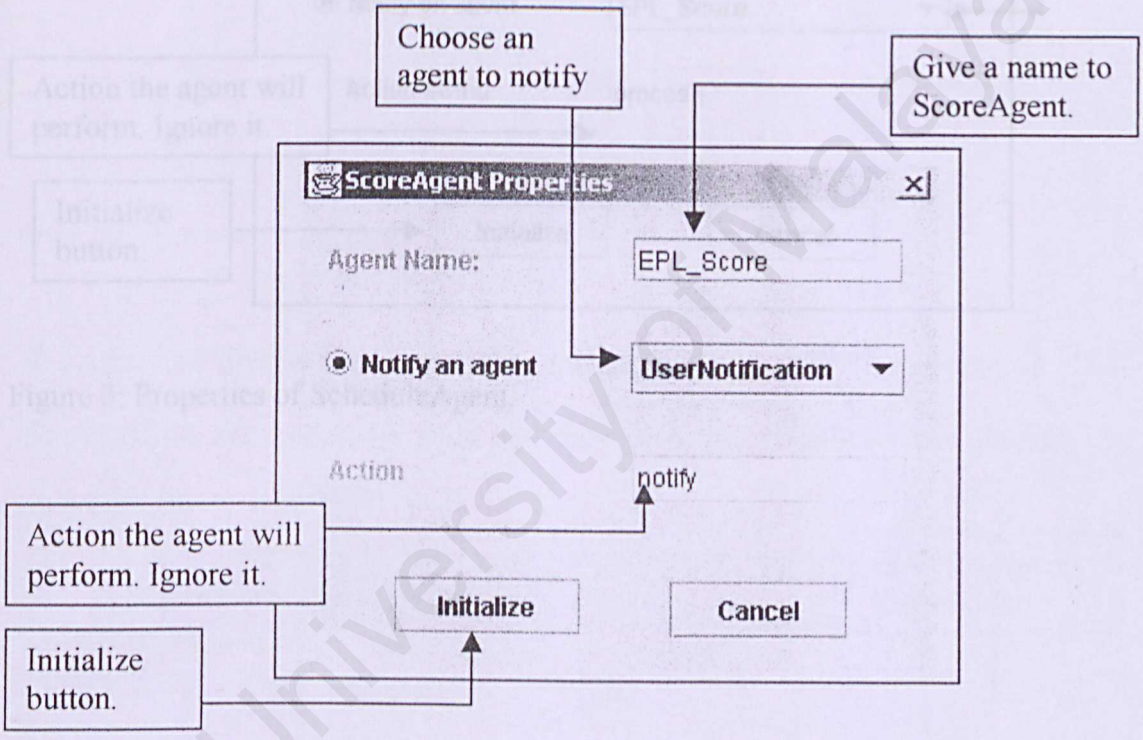


Figure 3: Properties of ScoreAgent.

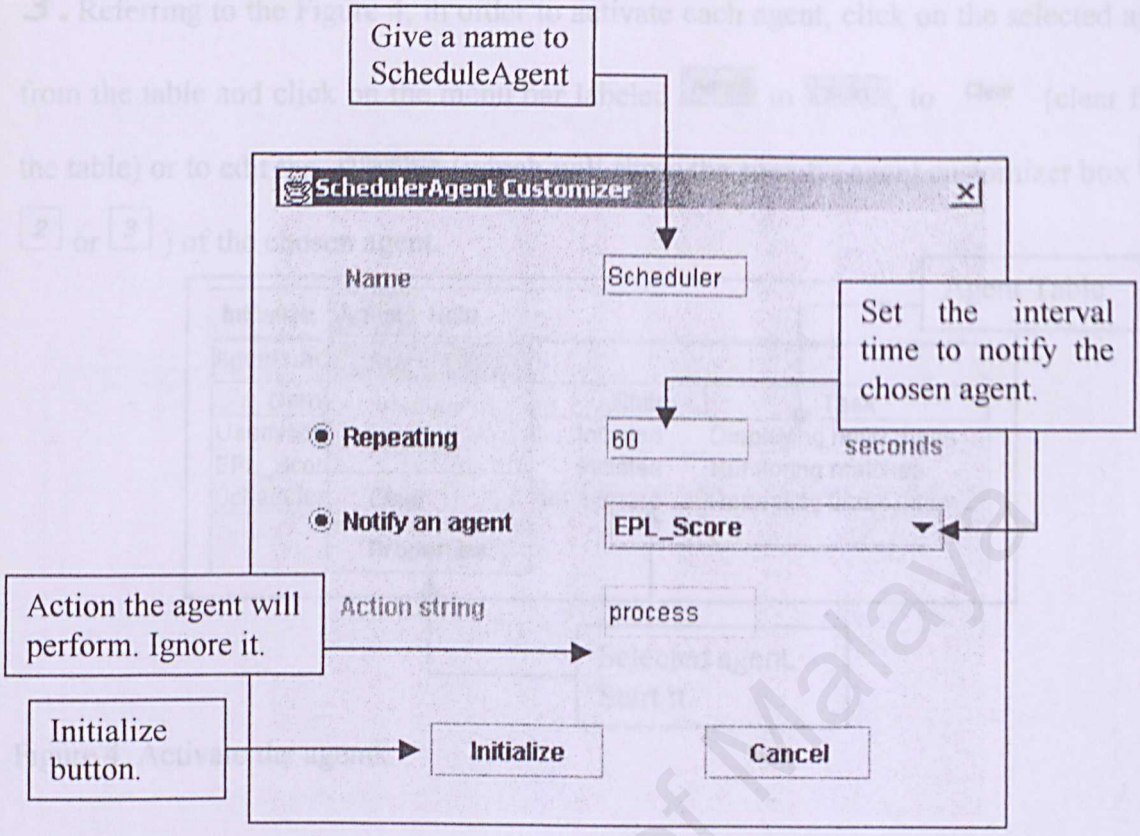


Figure 3: Properties of ScheduleAgent.

3. Referring to the Figure 4, in order to activate each agent, click on the selected agent from the table and click on the menu bar labeled **Agent** to **Start**, to **Clear** (clear from the table) or to edit the **Properties** (which will show the specific agent customizer box [1], [2] or [3]) of the chosen agent.

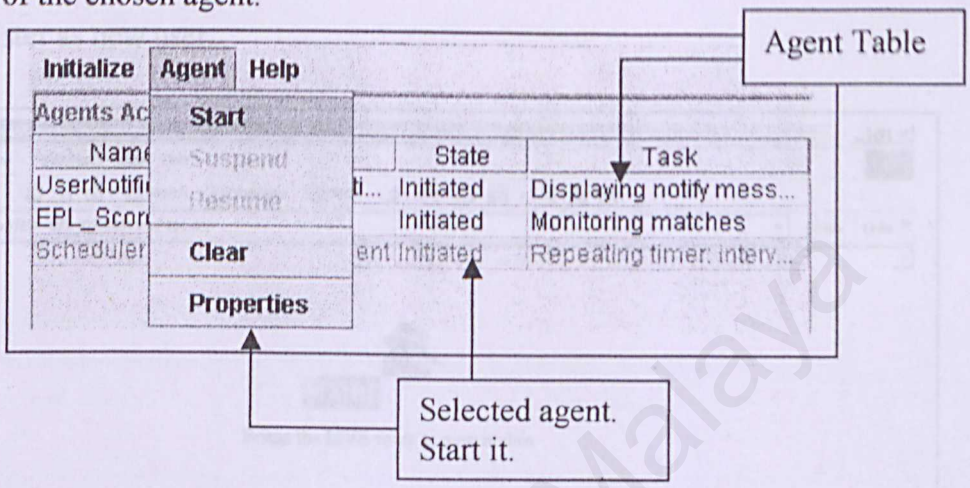


Figure 4: Activate the agents.

4. After the agents were activated, the menu item **Suspend** is enabled. This will deactivate the selected agent from performing its task. If this item is clicked, this will enable the menu item **Resume**.

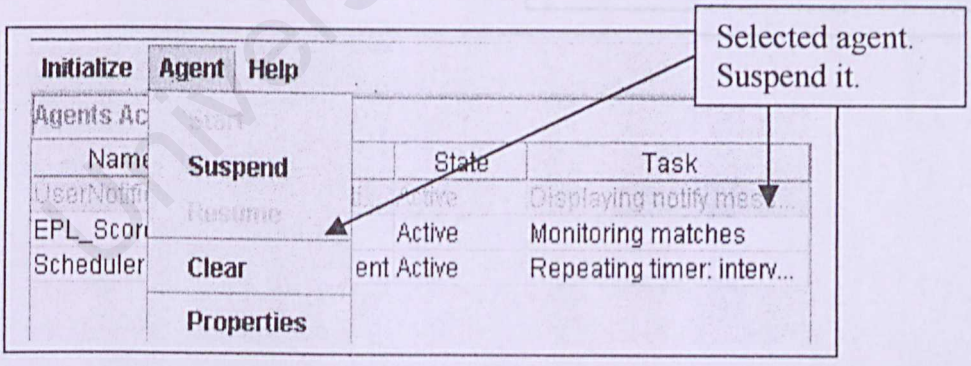


Figure 5: Agent's activities and properties.

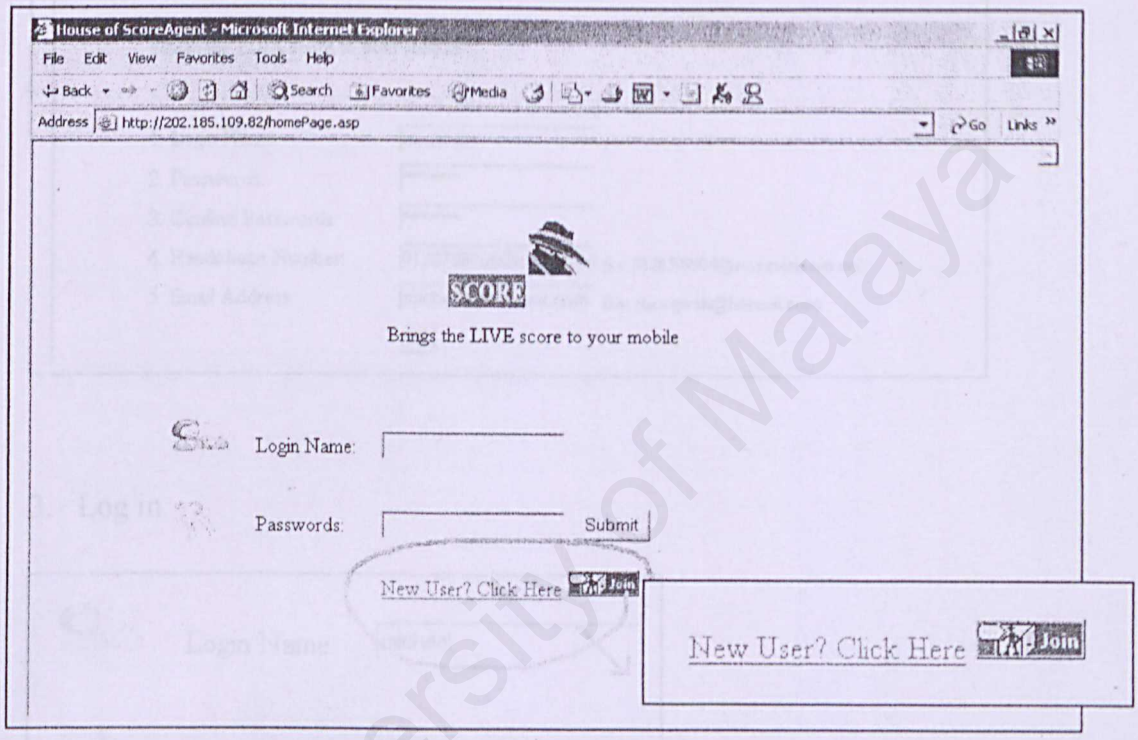
After, all the agents have started, the administrator can leave the tasks to agents. The agents will perform according to their designation.

3.0 USER REGISTRATION

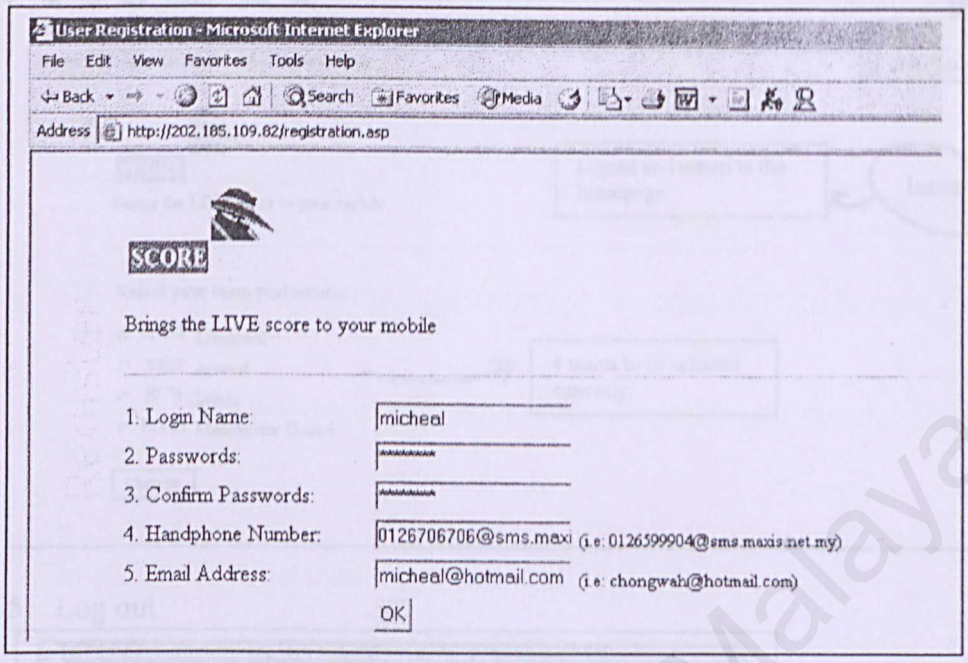
At the moment, user can log on to **http://202.185.109.82/homepage.asp**

Steps:

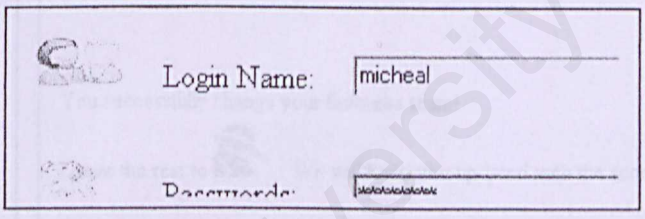
1. Register as new user.



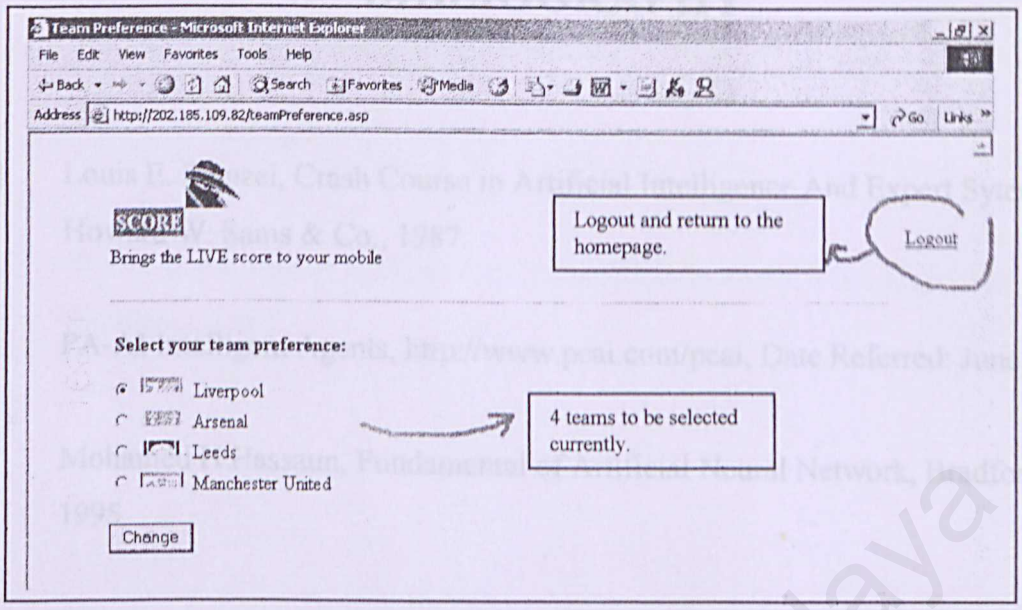
2. Input new user's information.



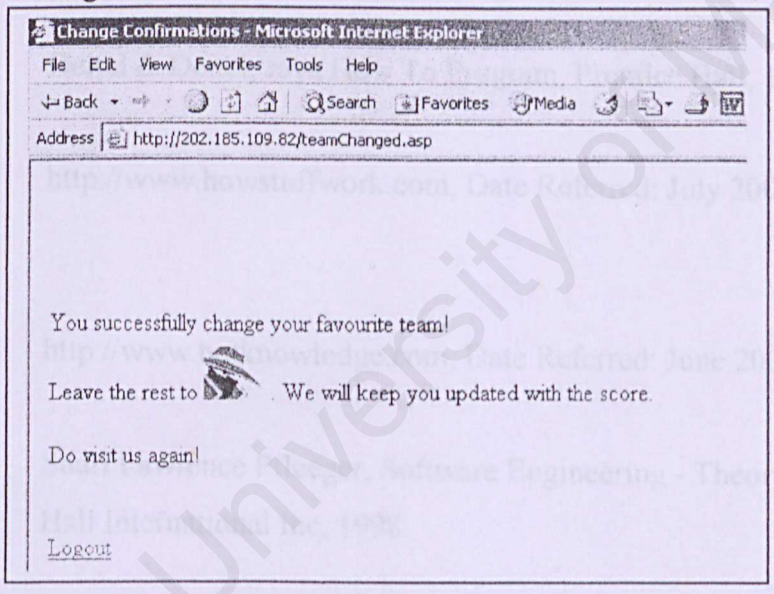
3. Log in.



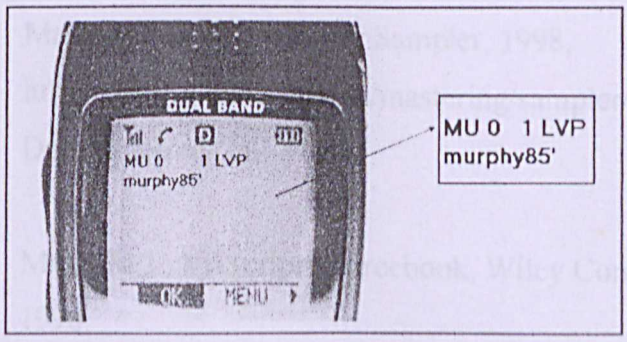
4. Select or change user's team preference.



5. Log out



4.0 THE ACTUAL OUTCOME



BIBLIOGRAPHY

[1] Louis E. Frenzel, Crash Course in Artificial Intelligence And Expert Sytem, Howard W. Sams & Co., 1987.

[2] PA-AI Intelligent Agents, http://www.pcai.com/pcai, Date Referred: June 2001.

[4] Mohamed H Hassaun, Fundamental of Artificial Neural Network, Bradford Book, 1995.

[5] http://www.whatis.com, Date Referred: June 2001.

[6] Deitel & Deitel, Java How To Program, Prentice Hall, 1999.

[7] http://www.howstuffwork.com, Date Referred: July 2001.

[8] http://www.botknowledge.com, Date Referred: June 2001.

[9] Shari Lawrence Pfleeger, Software Engineering - Theory and Pratice, Prentice Hall International Inc, 1998.

[10] http://www.microsoft.com, Date Referred: July 2001.

[11] Mastering Series Training Sampler, 1998, http://www.microsoft.com/mastering/sampler/mdad/mv600209.htm, Date Referred: July 2001.

[12] Mara, M.J., VBScript Sourcebook, Wiley Company Publishing, Indianapolis, 1996.

[13] Homer, A., Professional ASP, Bermingham, Wrox Press, 1997.

[14] Castro, E., HTML for World Wode Web, New York, Peachpit Press, 1996.

[15] John Durkin, Expert Systems Design and Development, Macmillan Publishing Company, 1994.