A MULTIMEDIA-ENHANCED EXPERT SYSTEM FOR JAVA PROGRAMMING

by

SITI ZUBAIDAH JERAI

FINAL PROJECT REPORT

Submitted to the Information Technology Programme
in Partial Fulfillment of the Requirements
for the Degree
Bachelor of Technology (Hons)
(Information Technology)

Universiti Teknologi Petronas
Bandar Seri Iskandar
31750 Tronoh
Perak Darul Ridzuan

CERTIFICATION OF APPROVAL

A MULTIMEDIA-ENHANCED EXPERT SYSTEM FOR JAVA PROGRAMMING

by

Siti Zubaidah Jerai

A project dissertation submitted to the
Information Technology Programme
Universiti Teknologi PETRONAS
in partial fulfilment of the requirement for the
Bachelor of Technology (Hons)
(Information Technology)

Approved:

Mr. Nordin Zakaria
Project Supervisor

Mohamed Nordin Zakaria
Lecturer
Information Technology/Information Systems
Universiti Teknologi PETRONAS
Bandar Seri Iskandar
31750 Tronoh
Perak Darul Ridzuan, MALAYSIA.

UNIVERSITI TEKNOLOGI PETRONAS
TRONOH, PERAK

June 2005

TABLE OF CONTENTS

CERTIFICATION OF ORIGINALITY

This is to certify that I am responsible for the work submitted in this project, that the original work is my own except as specified in the references and acknowledgements, and that the original work contained herein have not been undertaken or done by unspecified sources or persons.

SITI ZUBAIDAH JERAI

ABSTRACT

Final year project (FYP) paper is a compulsory to all UTP final year students. I am doing a FYP project titled "A Multimedia-Enhanced Expert System for Java Programming". This project focuses on developing an expert system which will be used in helping students solve error problems in Java Programming language. This report focus on the development of expert system, cover why it is developed, who need this system, when to use it, what are the problems occur during development, how the problems are solved and finally how the system benefit users at the end. The objective of this project is to come out with the Expert System for Java Programming. Recently, students are using the BlueJ application to compile a Java program and Java 2 SDK to execute the program. If some errors occurred during compiling and executing the program, it must be solved first to get the expected output. The problems here, students find that it is sometimes difficult for them to solved the errors occurred by only depending to the compiler. The compiler only gives the error messages. This system is developed and be expected to guide user in finding the correct solution for any errors occurred in Java programs. Targeted user for this system is UTP students. Throughout this report, the development and implementation of this project is deeply explained and elaborated.

ACKNOWLEDGEMENT

I am very pleased and satisfied with this Final Year would like to thank the following persons for sharing their unique talents and supports.

I begins my expressions of gratitude to the backbone Of FYP Committee – Mr. Mohammad Noor Ibrahim and Ms Vivien, and all IT/IS lecturers, (for giving full commitment in term of providing info about the final year project) Special gratitude to his supervisor, Mr. Mohammad Noor Ibrahim, quite fortunate to have such high caliber superior. Thank you, Mr. Mohammad Noor for his strong leadership and support. His commitment and positive feedback has driven me to continuously improving the project. Special thanks go to my supervisor Mr. Mohd. Nordin for his supervision and guidance, I have come to rely upon his guidance in completing this project. Sincere thanks also go to all UTP staffs that did support the author in finishing the project.

Thanks message also goes to my mother, Pn Zaliha Atan and my whole family who supports me financially and mentally.

And finally I also would like to thank my project mate and all my friend, who did contribute directly or indirectly towards my project . The presence of them during provided the author with the much-needed breaks and always, inspiration.

Thank you.

LIST OF FIGURES

LIST OF APPENDIX

CHAPTER 1
INTRODUCTION

1. INTRODUCTION

1.1 Background of Study

An expert system is a class of computer programs in artificial intelligence (AI). AI aims to build computersand programs so that they imitate human expertise in solving problems. The computer programs are commonly known as Expert System, also called Knowledge-Based Expert System. Expert System is an information systsem that applies the reasoning process of a human expert to new situations for decision making. The discipline of building expert system is called knowledge engineering. Its involves the process of acquiring knowledge from human experts.

In essence, expert system are programs made up of a set of rules that analyze information that usually supplied by the user of the system. The information is about a specific class of problems, as well as provide analysis of the problems, and, depending upon their design, a recommended course of user action in order to implement corrections.

There are generally three individuals having an interaction with expert systems. Primary among these is the end-user, the individual who uses the system for its problem solving assistance. In the building and maintenance of the expert system there are two roles: 1) the problem domain expert who builds the knowledge base and 2) a knowledge engineer who assists the experts in determining the representation of their knowledge and who defines the inteference technique required to obtain useful problem solving activity.

Most expert systems are developed via specialized software tools called shells. These shells come equipped with an inference mechanism (backward chaining, forward chaining, or both), and require knowledge to be entered according to a specified format. Examples of some rule-based shells are Exsys, InstantTea, XpertRule KBS, G2, Guru, K-Vision, CLIPS, and JESS. Jess supports the development of rule-based expert systems which can be tightly coupled to code written in the powerful, portable Java language.

The expert system discussed in this project, namely the Expert System for Java Programming, is aimed at capturing expertise knowledge in the area of java programming language, making it available for students or user to find solutions of errors occurred in java programming language. With the advent of the World Wide Web (WWW) and computer technology, not only expert system have managed to achieve fairly high levels of performance in task areas which require a good deal of specialized knowledge and training, but it can be easily accessible for use. Exploitation of potential human intelligence such as their knowledge, reasoning, and decision making and learning has led to the development of software that can assist users in making decision for solutions without consulting the respective expertise.

1.2 Problem Statement

1.2.1 Problem Identification

Using BlueJ application, user can compile a java programs. When there were errors occur, an error message is displayed indicating that the user have made a mistake and describing what the likely cause of the error might be. Unfortunately, the error messages are often difficult to interpret and are sometimes misleading.

A syntax error occurs when the codes violated one or more grammar rules of Java. Such errors are detected and displayed by the compiler as it attempts to translate the programs. If a statement has a syntax error, it cannot be translated and the program cannot be executed. Finding and removing syntax errors, one at a time, can be time consuming. To run a java program, user needs to install Java 2 SDK and type some command. If there is execution error which cause to execution failure, user has to check for the directory, the .txt file and make sure everything is correct. With only the error message displayed, users find it is hard for them to make correction and to find the portion of code which caused the error.

1.2.2 Significance of the Project

A Multimedia-Enhanced Expert System for Java Programming is designed to improve the process of correcting errors after debugging a program. This system is designed to help and guide user to correct errors accurately and reduce time consume in correcting errors after debugging a program. Expert systems make knowledge accessible to people who query the systems for advice. Expert systems can work faster than humans and that means fewer workers are needed. Therefore, it reduced the costs and increased the output.

Addressing problems requiring expertise involves addressing the two big issues of artificial intelligence: knowledge representation and search. The search space for a typical expert domain is very large. One of the characteristics of human experts is their efficiency in finding solutions to problems quickly and efficiently with the minimum of false tracks. To become expert in this way usually requires years of practical experience, during which the expert learns to associate characteristics of the problems to be solved with the most appropriate interpretation.

These associations comprise the expert's expertise. They enable the expert to go very quickly from a description of the problem to the solution: the diagnosis of a medical ailment based on the symptoms of the patient, for example; or the classification of a plant based on its botanical features, and so on.

By incorporating the expertise of the human expert into a computer program, we can reduce the size of the search space for solving problems. The expertise describes the valid inferences which can be deduced from the known problem features.

1.3 Objective and Scope of Study

1.3.1 Objectives

The objective and scope of work to be achieved are as follows:

1) To understand the underlying concepts of AI and expert system.

2) To do analysis and observations on common problems occur while using java programming language.

3) To examine the rule based expert system approach in constructing a computer based knowledge education for Java programming.

4) To apply theoretical knowledge by developing a good and successful expert system for Java programming. This expert system can be used as a better alternative on correcting errors in Java than the previous version of compiler.

5) To develop an expert system for java programming, I must focus on solving common errors faced by students or users. The system will capture a programmer expert knowledge and make it available for users like students to, who need a solution for a problem faced. The expert system is aimed at serving as a useful assistant in guiding user for the correct solutions without consulting their lecturer or other human expert. By giving a few inputs on the related error message displayed, a non-expert will be able to obtain results and solution for the possible problems.

1.3.2 Scope of Study

This will focus on development of an expert system for Java programming to guide for finding solution for any errors occur in java programs, as how it would be undertaken by a human expert like lecturer.

This project will focus on developing an expert system that poses questions on errors that user are facing, to user, and based on the input provided, will make a decision on what the user should do next. The system mimics the way a human expert would guide a student to correct the coding errors. Emphasis on what action should be taken next will be given to knowledge acquisition to develop the knowledge base and inference methods to deduce the solutions.

1.3.3 Feasibility of the Project within the Time and Scope

This project can be deemed as technically feasible as the scope of the project is limited to the solving common errors occur in Java programming. There is no relative cost related to the project as the expert system can be developed using software that is already available in the Internet. There are also adequate resources available to support the project, such as books, online resources, lecturers and students.

The time frame given to complete the project is also sufficient. The Gantt chart produced indicates the time allocated for each task it serves as guidance for project execution.

CHAPTER 2
LITERATURE REVIEW AND THEORY

2. LITERATURE REVIEW AND THEORY

2.1 Expert System

An expert system is a class of computer programs developed by researchers in artificial intelligence during the 1970s and applied commercially throughout the 1980s. In essence, they are programs made up of a set of rules that analyze information (usually supplied by the user of the system) about a specific class of problems, as well as provide analysis of the problem(s), and, depending upon their design, a recommended course of user action in order to implement corrections.

Expert systems are computer programs that carry out reasoning or problem solving tasks in a specific domain of application. Typical examples are systems that perform diagnosis in the domains of medicine or car mechanics; classification of plants, butterflies, etc.; even the design of rooms according to the principles of feng shui! They are called 'expert' systems because, as well as incorporating information about the domain of application, they include additional information about how to do the problem solving task; this information is obtained by asking or observing human experts solving similar tasks.

A related term is wizard (software). Like an expert system, a wizard is also an interactive computer program that helps a user solve a problem. Usually, the term wizard is used for programs that search a database for criteria entered by the user. Unfortunately, the distinction between these two definitions is not universal, and some rule-based programs are called wizards.

Type of problems solved by expert systems

Typically, the problems to be solved are of the sort that would normally be tackled by a human "expert" - a medical or other professional, in most cases. Real experts in the problem domain (which will typically be very narrow, for instance "diagnosing skin diseases in human teenagers") are asked to provide "rules of thumb" on how they evaluate the problems, either explicitly with the aid of experienced system developers, or sometimes implicitly, by getting such experts to evaluate test cases and using computer programs to examine the test data and (in a strictly limited manner) derive rules from that. Simple systems use simple true/false logic to evaluate data, but more sophisticated systems are capable of performing at least some evaluation taking into account real-world uncertainties, using such methods as fuzzy logic. Such sophistication is difficult to develop and still highly imperfect.

Application

While expert systems have distinguished themselves in AI research in finding practical application, their application has been limited. Expert systems are notoriously narrow in their domain of knowledge—as an amusing example, a researcher used the "skin disease" expert system to diagnose his rust bucket car as likely to have developed measles—and thus prone to making errors that humans would easily spot.

Additionally, once some of the mystique had worn off, most programmers realized that simple expert systems were essentially just slightly more elaborate versions of the decision logic they had already been using. Therefore, some of the techniques of expert systems can now be found in most complex programs without any fuss about them.

Expert systems versus problem-solving systems

The principal distinction between expert systems and traditional problem solving programs is the way in which the problem related expertise is coded. In traditional applications, problem expertise is encoded in both program and data structures. In the expert system approach all of the problem related expertise is encoded in data structures only. None is in programs. Several benefits immediately follow from this organization.

An example may help contrast the traditional problem solving program with the expert system approach. The example is the problem of tax advice. In the traditional approach data structures describe the taxpayer and tax tables, and a program in which there are statements representing an expert tax consultant's knowledge, such as statements which relate information about the taxpayer to tax table choices. It is this representation of the tax expert's knowledge that is difficult for the tax expert to understand or modify.

In the expert system approach, the information about taxpayers and tax computations is again found in data structures, but now the knowledge describing the relationships between them is encoded in data structures as well. The programs of an expert system are independent of the problem domain (taxes) and serve to process the data structures without regard to the nature of the problem area they describe. For example, there are programs to acquire the described data values through user interaction, programs to represent and process special organizations of description, and programs to process the declarations that represent semantic relationships within the problem domain and an algorithm to control the processing sequence and focus.

The general architecture of an expert system involves two principal components: a problem dependent set of data declarations called the knowledge base or rule base, and a problem independent (although highly data structure dependent) program which is called the inference engine.

Individuals Involved with Expert Systems

There are generally three individuals having an interaction with expert systems. Primary among these is the end-user; the individual who uses the system for its problem solving assistance. In the building and maintenance of the system there are two other roles: the problem domain expert who builds the knowledge base, and a knowledge engineer who assists the experts in determining the representation of their knowledge and who defines the inference technique required to obtain useful problem solving activity.

The Knowledge Engineer

The knowledge engineer is concerned with the representation chosen for the expert's knowledge declarations and with the inference engine used to process that knowledge. There are several characteristics known to be appropriate to a good inference technique.

1. A good inference technique is independent of the problem domain.
   In order to realize the benefits of explanation, knowledge transparency, and reusability of the programs in a new problem domain, the inference engine must not contain domain specific expertise.

2. Inference techniques may be specific to a particular task, such as diagnosis of hardware configuration. Other techniques may be committed only to a particular processing technique.

3. Inference techniques are always specific to the knowledge structures.

4. Successful examples of rule processing techniques include:
   (a) Forward chaining
   (b) Backward chaining

The Inference Rule

An understanding of the "inference rule" concept is important to understand expert systems. An inference rule is a statement that has two parts, an if-clause and a then-clause. An expert system's rule base is made up of many such inference rules. They are entered as separate rules and it is the inference engine that uses them together to draw conclusions. Because each rule is a unit, rules may be deleted or added without affecting other rules (though it should affect which conclusions are reached). One advantage of inference rules over traditional programming is that inference rules use reasoning which more closely resemble human reasoning.

Thus, when a conclusion is drawn, it is possible to understand how this conclusion was reached. Furthermore, because the expert system uses knowledge in a form similar to the expert, it may be easier to retrieve this information from the expert.

The User Interface

The function of the user interface is to present questions and information to the operator and supply the operator's responses to the inference engine. Any values entered by the user must be received and interpreted by the user interface. Some responses are restricted to a set of possible legal answers, others are not.

The user interface checks all responses to insure that they are of the correct data type. Any responses that are restricted to a legal set of answers are compared against these legal answers. Whenever the user enters an illegal answer, the user interface informs the user that his answer was invalid and prompts him to correct it. As explained in the cross" referenced application, communication between the user interface and the Inference Engine is performed through the use of a User Interface Control Block (UICB) which is passed between the two.

Benefits of Expert Systems

- Increased output & productivity - Expert systems can work faster than humans and that means fewer workers are needed. Therefore, it reduced the costs and increased the output.
- Reduced downtime - Expert systems can save a considerable amount of money for the company involved by reducing the downtime.
- Increased quality - Expert systems provides consistent advice and reduces the rate of error.
- Capture of scarce expertise - The scarcity of expertise becomes evident in situations where the expert is retiring or leaving the job.
- Reliability - Expert systems are reliable as they do not take medical leave, go on strike, or get tired.
- Accessibility to knowledge & help desks - Expert systems make knowledge accessible to people who query the systems for advice.
- Increased capabilities of other computerized systems

2.2 Tools available for the System Engine

CLIPS

CLIPS is freely downloadable from the CLIPS internet-site http://www.ghg.net/clips for non-commercial use. The package is downloadable as a collection of source-files or as an executable from the web page. CLIPS is also included with the book Expert Systems: Principles and Programming, 3rd Edition, by Giarratano and Riley and a Reference Manual. After uncompressing the file, the executable can be started and the user is immediately confronted with the CLIPS environment. The User's manual, together with this environment, makes it easy to begin programming and experimenting with the software.

JESS

The JESS package can be obtained on the URL http://herzberg.ca.sandia.gov/jess, free from charge (for non-commercial use and for commercial use within the U.S.) as well. There are two possibilities: either JESS for UNIX or JESS for MS Windows can be downloaded. The former is compressed in tar's format, the latter in .zip format. In contrast with CLIPS, after the JESS distribution packages have been uncompressed, the whole software must be compiled from .java into .class files. This compiling procedure has a few tricky edges: the compile command must be executed from the main JESS directory; otherwise the Java compiler will give all kinds of errors. Furthermore, the right version of the Java compiler and virtual machine must be used. The included readme.html file gives only a brief (and incomplete) description on this procedure, so inexperienced users will need some time and patience.

EXSYS CORVID

Exsys CORVID provides a powerful environment for developing Web-enabled Knowledge Automation Expert Systems for a wide range of decision-making problems. CORVID allows the logical rules and procedural steps used by an expert to make a decision, to be efficiently described in way that is easy to read, understand and maintain. CORVID converts that logic to a form that the underlying Runtime Inference Engine can process to emulate the questions, process, and recommendations of the expert in an interactive session that can be delivered over the Web. This allows end users to interact with the system as if they were talking to the expert, to produce situation-specific recommendations and advice on a wide range of subjects.

2.3 Java Technology - The Key Differentiator

As a leader in delivering technology education to IT professionals, Sun Educational Services trains more than 120,000 students a year in more than 52 countries. Sun Educational Services assists businesses and IT professionals with high-quality education and Sun certification, including Java technologies. Java technology is quickly becoming a standard for IT application development, as shown by recent statistics.

"Java is now the fastest-growing programming environment (language and platform) in the world," says Mark Driver, research director of the Gartner Group, in a recent research note, "Where are Java Programmers When You Need Them?" Sun has had more than six million downloads of Java 2 Enterprise Edition (J2EE) and Java Developers Kit 1.1 in a recent five-month period.

The tremendous growth of Java technologies has resulted in a skyrocketing demand for programmers, developers and software architects. Sun estimates the number of programmers on the Java platform at approximately 1.5 million worldwide. Analysts predict further growth in demand for skilled Java technology professionals, who have bright prospects for career advancement.

"There is a significant shortfall between the number of developers needed to meet the demand for new Java-based projects and the actual number of qualified resources available to fill these positions," according to Driver. This skills gap is affecting not only mainstream businesses, but also consulting and IT outsourcing organizations.

Driver has predicted that "by 2003, 60 percent of large enterprises will use externally sourced workers to fulfill half of their IT activities." Driver has also said that "the Java skills gap will be a contributing factor in this trend."

Companies are seeking qualified, effective IT professionals to integrate Java technology into their business- and mission-critical IT applications. How does a company determine if an individual is qualified? Employers are increasingly using certification to conveniently validate the competence of IT professionals with Java technology experience.

As an industry standard, certification gives businesses confidence that an IT professional has a certain skill set. Service providers typically prefer to hire individuals with Java certification to prove that they have the expertise to develop and support Java technology. Certification can also bring peace of mind to IT managers. Several studies over the past five years have shown that certified staff make fewer errors, incur lower support costs and are typically more productive than their non-certified peers.

Similarly, certification helps individuals demonstrate that they have the knowledge to do the job correctly and justify higher compensation and promotions. Sun-certified professionals can rapidly establish credibility, resulting in peer recognition and professional advancement. As one recent Sun-certified professional said, "Certification helped cement my position in the Java technology community. It helps me get good Java work, and it gives me better credentials with my employer."

2.4 Java Development Trends Emerging

No-strings-attached Java

Wireless application projects written in Java are picking up steam. This is happening largely because Nokia, the world's largest mobile telephone maker, has committed to using the open source Symbian OS for all of its future handsets. Java apps of all kinds—from heavy-duty enterprise CRM apps to simple video games—run wonderfully on it. And Nokia competitor Qualcomm's homemade operating system, BREW, also runs Java applications smoothly for another whole fleet of handheld devices.

Real-time Specification for Java

The Real-time Specification for Java (RTSJ) is coming into its own. This is where Java creator James Gosling and Sun's real-time Java guru, Greg Bollella, are spending much of their professional time right now: strategizing and promoting real-time Java apps for embedded use.

It's beginning to pay off. Gosling and Bollella have described embedded systems as "the new frontier in which predictable execution takes precedence" over other system attributes, such as speed, bandwidth, and payload-carrying ability. (See Sun's Real-time Specification for Java for more detailed information.) The RTSJ addresses language and runtime issues that cause unpredictability. The JCP is debating and editing the specification now; it still has a way to go (most likely several months) before final approval and implementation.

CHAPTER 3
METHODOLOGY

3. METHODOLOGY

3.1 Procedure Identification

For this project, the development methodology used is the expert system development life cycle (ESDLC) *(see Figure3.1)* that incorporates prototyping during its system development. The ESDLC begins with a problem. The knowledge engineer (students) works with the experts (lecturer) to identify the problem domain and then builds the expert system.

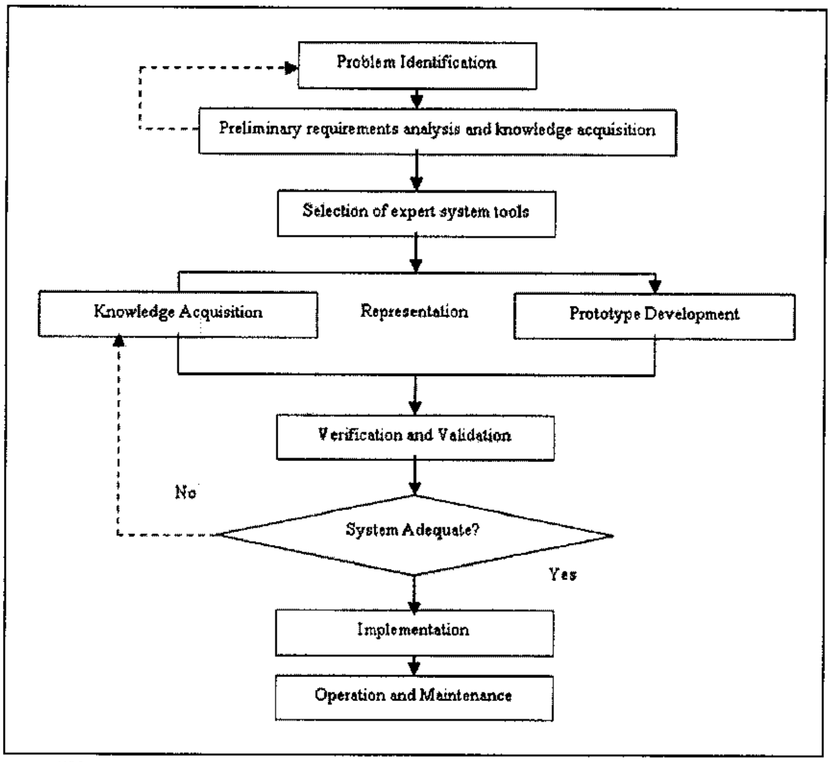

Problem Identification
Preliminary requirements analysis and knowledge acquisition
Selection of expert system tools
Knowledge Acquisition
Representation
Prototype Development
Verification and Validation
No
System Adequate?
Yes
Implementation
Operation and Maintenance

Figure 3.1: Expert System Development Life Cycle

Knowledge acquisition is the initial phase whereby knowledge is acquired from the experts. Knowledge is then organized and represented by rules contained in the system knowledge based. The prototype of the system is developed and taken on for verification and validation. Verification and validation involves exposing the prototype to user comment and refining it through many stages or versions until adequate system has been developed. Expert system development involves heavy user involvement with the development process to ensure validity and correctness of the system.

The knowledge acquisition begins with a literature study on the expert system and the observations done to students which are taking Object Oriented Programming course. With a literature review and observations, questionnaire to elicit expertise opinion are prepared. Interviews are conducted with multiple experts (lecturers and tutors) to seek their input and knowledge. The collected information is reviewed and analyzed. This process provides an insight into how knowledge will be organized and represented, and problem solving concepts established. This knowledge, organized in a form of decision tree is transformed into production rules in the knowledge base, using Exsys CORVID. Knowledge acquisition in an evolving step as verification and validation would require correcting and updating the knowledge base.

The knowledge obtained for this Java Programming Expert System is done through the knowledge acquisition process as shown in Figure 3.2.

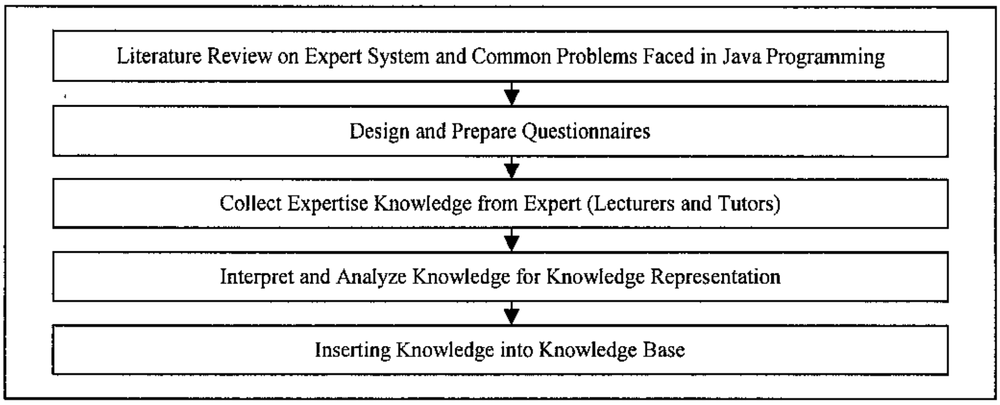

Literature Review on Expert System and Common Problems Faced in Java Programming
Design and Prepare Questionnaires
Collect Expertise Knowledge from Expert (Lecturers and Tutors)
Interpret and Analyze Knowledge for Knowledge Representation
Inserting Knowledge into Knowledge Base

Figure 3.2: Flowchart of Knowledge Acquisition Process

The design stage involves organizing and representing the expert's knowledge and information flow into formal representations. The program's logic is designed at this stage. The knowledge collected is grouped into their categories based on common errors faced while doing Java programming. The problem solving steps are developed and its tree structures are constructed for its knowledge base. The design stages involves determining the attributes (variable in Exsys CORVID) and which of these attributes should be asked to the user or represented by an internal set of decision tree.

The first step in this incremental decision-making process is to identify the type of errors occurred in the Java programs. After this initial decision, similar decision nodes follow in order to arrive to the expected and correct result. These decisions depend upon the error messages appeared on the compiler.

The actual working code of this system is represented by rules. Based on the decision that is constructed, knowledge is extracted from the gathered information, and then inserted into a knowledge base of the system and in the form of production rules. A production rules specifies action to be taken if a certain condition of errors occurred in a Java program. Built into the system's knowledge base is a list of errors and common problems faced by the users which perform Java programming. The Inference Engine match rules against the current state of the system, which are the facts. The user interface design is also built at this stage to determine the user interaction with the system.

An example of the production rules implemented in this expert system is as follows:

Rule
IF the errors type = compilation errors
AND error message displayed = ';' expected
....
THEN put ';' at the end of the row

In Exsys CORVID, each premise and condition contains attributes and values. Attributes are called variables in Exsys CORVID. Variable are building blocks that are elements incorporated into the problem solving process. Variables are attribute-value pairs (AV pairs) that are used to define the logic in the Logic Blocks and Command Blocks of Exsys, to hold data during execution of the system and to define the goals of how the system will run. The AV pair is the fundamental building blocks of a premise or conclusion, hence the production rules. In the rule above, the attribute is the errors type, and the value be tested against it is a compilation errors. Each AV pair is associated with a set of properties as follows:

Name
: Name of the attribute.
(ex: errors type-like compilation or execution errors)

Type
: Attribute/variable value may be numeric, string, static list or symbolic. Some variables are specified as static list, which are multiple choice list with the values defined during the development of the system. Below shows attribute = error, with the list of answers.

Example:

What type of error does occur on your program?

a. Compilation errors
b. Execution errors

Prompt
: The query presented to the user when asking user for input or in displaying results. The user has to reply to this prompt. For example a user may be asked like the above example.

Confidence Factors
: Confidence Factors (CF) is associated with the AV or values. The goals of this system include a CF for a particular solution or action to be taken.

3.2 Tools Required

For the hardware part, personal computer with processing speed of 1.5GHz, 256MB RAM and hard disk storage of 20GB should be sufficient in developing this system.

I have listed and briefly explained all the software been used in developing this system:

- Windows 98/ME/2000/XP
    - I used windows XP as the operating system for my PC

- BlueJ application
    - This is the recent compiler used by most students. I installed this compiler to experience myself on using it to compile a Java program.

- Java 2 JDK
    - I used the latest Java Software Development Kit which is j2sdk1.4.2 to test and run Java program.

- Exsys Corvid
    - I chose to use Exsys CORVID to develop this system because Exsys CORVID provides a powerful environment for developing Web-enabled Knowledge Automation Expert Systems for a wide range of decision-making problems.

- Macromedia DreamWeaver MX
    - I used Macromedia Dreamweaver MX to design and edit the interface of this system.

3.3 System Architecture

Figure 3.3 shows the most important modules that make up a rule-based expert system. The user interacts with the system through a *user interface* which may use menus, natural language or any other style of interaction). Then an *inference engine* is used to reason with both the *expert knowledge* (extracted from our friendly expert) and data specific to the particular problem being solved. The expert knowledge will typically be in the form of a set of IF-THEN rules. The *case specific data* includes both data provided by the user and partial conclusions (along with certainty measures) based on this data. In a simple forward chaining rule-based system the case specific data will be the elements in *working memory.*

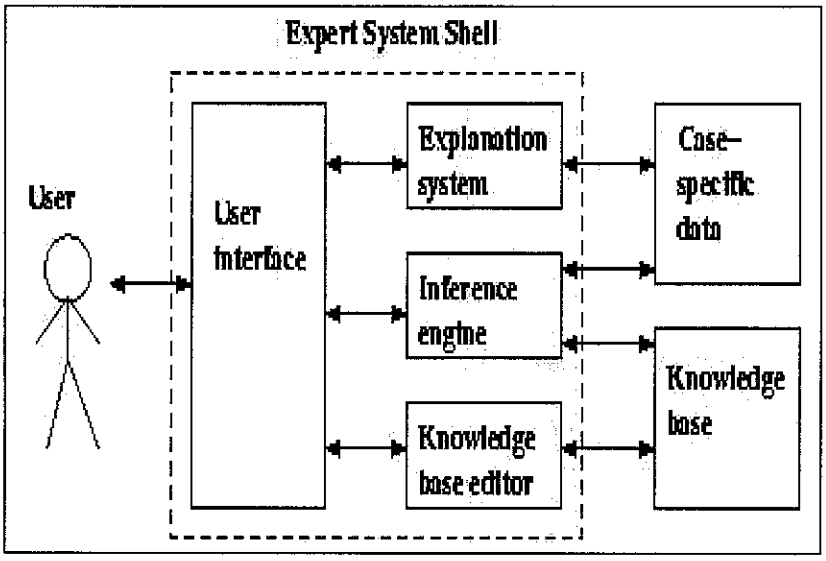


Figure 3.3: Expert System Architecture for Java Programming Expert System

One important feature of expert systems is the way they (usually) separate domain specific knowledge from more general purpose reasoning and representation techniques. The general purpose bit (in the dotted box in the figure) is referred to as an *expert system shell.* As we see in the figure, the shell will provide the inference engine (and knowledge representation scheme), a user interface, an explanation system and sometimes a knowledge base editor. Using shells to write expert systems generally greatly reduces the cost and time of development (compared with writing the expert system from scratch).

The *Expert System Shell* use to develop Expert System for Java Programming is Exsys CORVID.

The *user*, who is the students, will use the expert system as problem solving aid.

The *knowledge base* is the repository for rules and facts. Facts are knowledge specific to domain of application known to prior to the consultation session between the user and the expert system. Rules are heuristics that describe relations or phenomena in the domain for problem-solving. Expert System for Java Programming is a rule-based system, with its knowledge represented in the form of IF...THEN rules. The Logic Block of Exsys CORVID contains these production rules.

The *inference engine* performs primary tasks. Firstly, it examines the status of the knowledge base to identify what facts are known at any given time and to add any new facts that become available. Secondly, it determines the order in which inferences are made. Exsys CORVID has the CORVID Inference Engine which is able to match rules against the current state of the system to apply the corresponding actions.

The *knowledge base editor* enters rules specified by the knowledge engineer, into the knowledge base during development phase of system.

The *user interface* handles all input to the computer and formats all output. Expert System for Java Programming user-friendly interface was built using the JAVA CORVID Runtime Applet and Interface Command Builders. User inputs data through the HTML Web Page, which links to the knowledge base (Logic Block), feeding the user input data. The knowledge base is consulted and decisions are made concerning the next input question to display to the user. The user answers the questions and this process is repeated until the CORVID Inference Engine comes to a final result, which is the solution for the problems occur in a Java program.

3.4 Prototype Development

I have prepared the rules earlier for the expert system. The rules are converted into the specific language used by the software and coded into the framework of the development tools to build a working prototype. The knowledge base and inference engine are established in this stage. Exsys CORVID is the software tool chosen for the implementation of this project. Prototype in system development involves creating a version or part of the system. Prototyping helps to verify the system's integrity and reliability.

3.4.1 Formalization of Knowledge

The real forte of expert system is their capacity to make inferences or the drawing conclusions from premise. This is precisely what makes and ES intelligent. Even when it is possible to represent domain knowledge as rules, a human expert would not only have to know how to apply these rules but in which order they should be applied to solve particular problem.

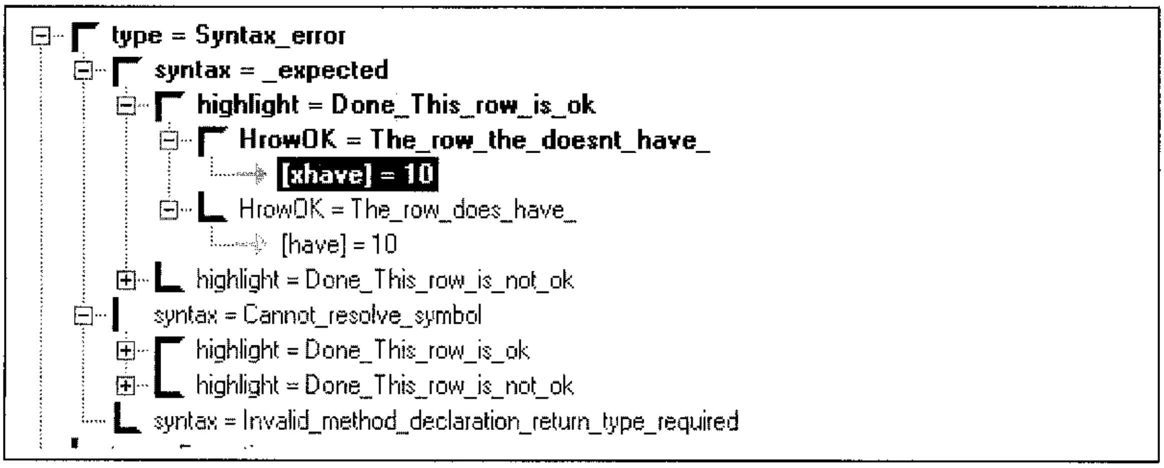
In Expert System for Java Programming, I have sketched the rules allow the overall problems in Java programming to be solved. The IF/THEN rule has proven to be the most effective and efficient way to describe the heuristics for the problem-solving process. In this system production rules, the IF past is tested to be true or false user input. If the truth of the IF part is confirmed, the statements THEN part will considered true. Below shows the declarative knowledge of Java Programming Expert System represented in the form of production rules and its corresponding rules represented in Exsys CORVID's Logic Block. The Logic Block is the knowledge base of this system.

Declarative knowledge of Java Programming Expert System in the form of production rules follow:

Example:

IF there is syntax error occurred
AND the error message displayed is ';'expected
AND the highlighted row is ok
AND the coding above the highlighted row doesn't have ';'
THEN put ';' at the end of the row

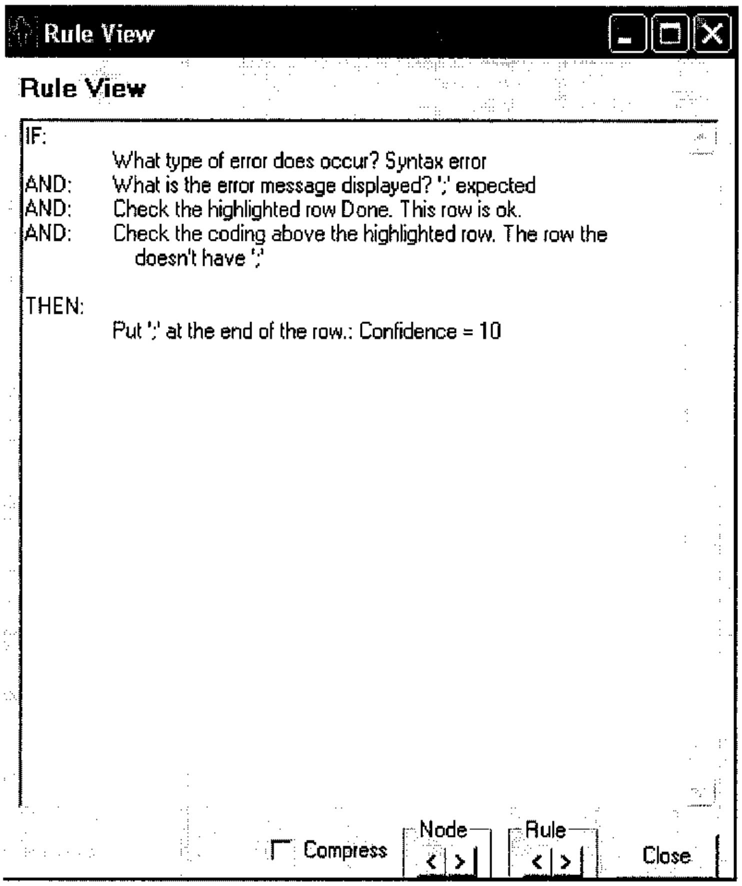
Using Exsys CORVID's Logic Block, this rule is expressed as follows:



```
type = Syntax_error
  syntax = _expected
    highlight = Done_This_row_is_ok
      HrowOK = The_row_the_doesnt_have_
        [xhave] = 10
      HrowOK = The_row_does_have_
        [have] = 10
    highlight = Done_This_row_is_not_ok
  syntax = Cannot_resolve_symbol
    highlight = Done_This_row_is_ok
    highlight = Done_This_row_is_not_ok
  syntax = Invalid_method_declaration_return_type_required
```

Figure 3.4: Part of Expert System for Java Programming Logic Block in Exsys CORVID

The Rule View in Exsys illustrates the full Variable prompt text as shown in Figure 3.5 as below:



Rule View

```
IF:
          What type of error does occur? Syntax error
AND:      What is the error message displayed? ';' expected
AND:      Check the highlighted row Done. This row is ok.
AND:      Check the coding above the highlighted row. The row the
            doesn't have ';'

THEN:
          Put ';' at the end of the row.: Confidence = 10
```

Figure 3.5: Rule View in Exsys CORVID

The rule in the Logic Block utilizes the AV-pairs, in which each rules has been named according to the attributes and their corresponding value. The THEN part is the action to be taken (solution) and it's associated CF. the CF indicates the solution of the problems occurred in the Java program based on the user input. The CF is associated with the validity of a knowledge base rule.

3.4.2 User Interface and System Integration

Exsys CORVID presents every question in a new page. User responds to the questions by selecting answer from a few options given. When there is no selected answer, the system will remain at the same page without proceeding, until user supplies the answer. The question pages are designed to be simple, providing clarity questions to users. Images are also embedded on some questions to provide an illustration of a particular answer user should give.

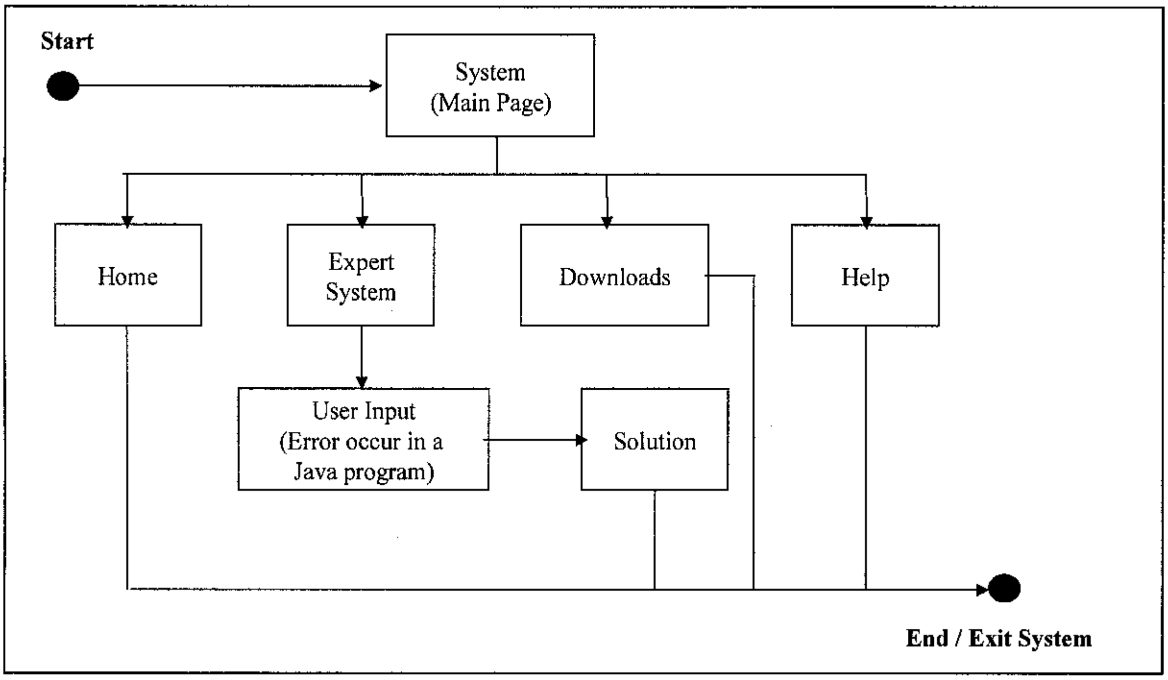


Figure 3.6: Java Programming Expert System Flow Chart from User Perspective

CHAPTER 4
RESULTS AND DISCUSSION

4. RESULTS AND DISCUSSION

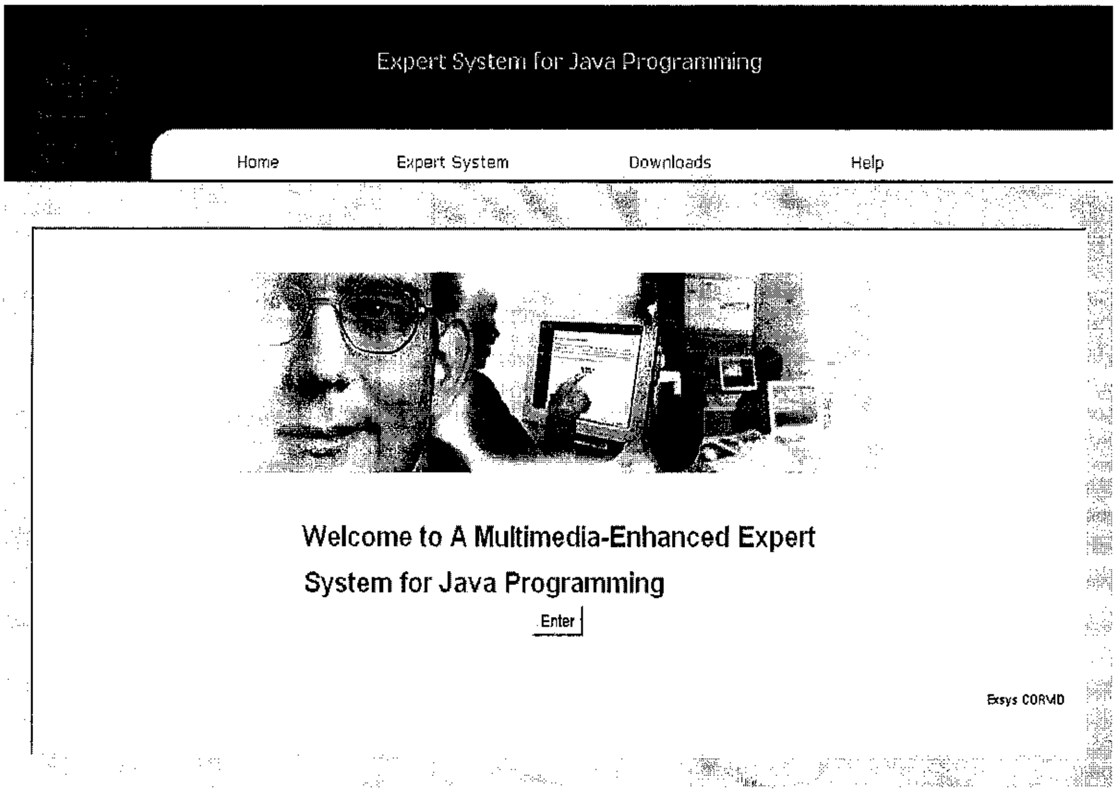
4.1 Results

The Expert System for Java Programming will capture a programmer expert knowledge and make it available for users like students to, who need a solution for a problem faced. The expert system is aimed at serving as a useful assistant in guiding user for the correct solutions without consulting their lecturer or other human expert. By giving a few inputs on the related error message displayed, a non-expert will be able to obtain results and solution for the possible problems.

Exsys CORVID has been chosen as the software tool to implement this system because of the development interface is easy to learn and understand. CORVID uses an "object-structure" approach to system design. Rules are defined using various types of variables that have associated methods and properties, providing a wide range of flexibility and power. Many of the advantages of a full object-oriented approach are provided without having to understand complex OO programming, or describing a solution with OO classes. For the expert system rules, the decision-making logic is stated in If-Then rules, in much the same way as explaining to another person how to make the decision. The rules are written in English and algebra, making them easy to read, understands and maintain.

Exsys Corvid provides tree structures to organize related rules, and Logic Blocks to organize related trees. CORVID provides many ways to make an expert system look great. The full design and functionality of HTML can be used for interface design.

Editable HTML templates make it easy to quickly design a consistent, professional interface. This expert system can be used as a better alternative on correcting errors in Java than the old version of compiler.

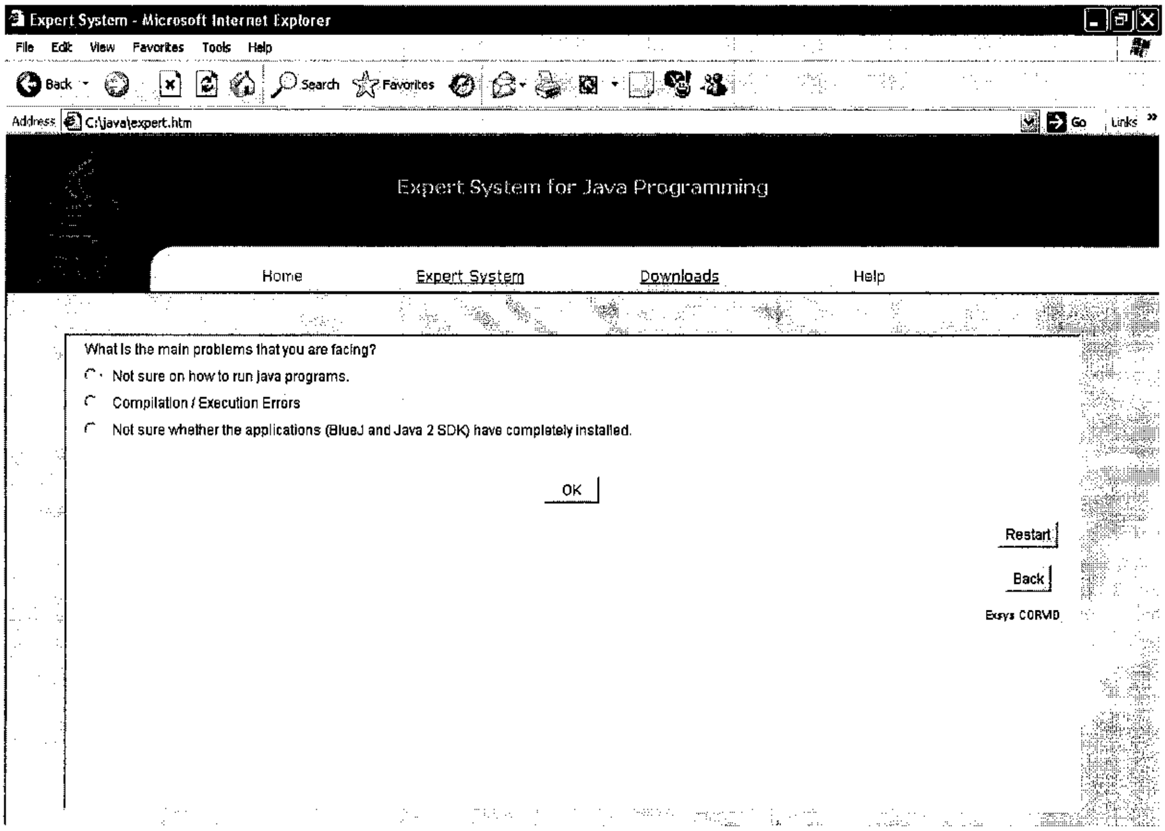


Welcome to A Multimedia-Enhanced Expert System for Java Programming

Figure 4.1: Main Page of Expert System for Java Programming

The main page shown in Figure 4.1 welcomes user into the system. Users can click the Enter button to begin using the system. User also can download the available installers provided on that system by clicking the 'Download' menu.

After a user has enter to the system and ready to use the system, they have to answer the first question appear on the screen and the following question until they get the final result. Figure 4.2 shows the initial questions that a user must answer.


Expert System - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back Search Favorites
Address C:\java\expert.htm
Go Links
Expert System for Java Programming
Home
Expert System
Downloads
Help
OK
Restart
Back
Exsys CORVID

What is the main problems that you are facing?

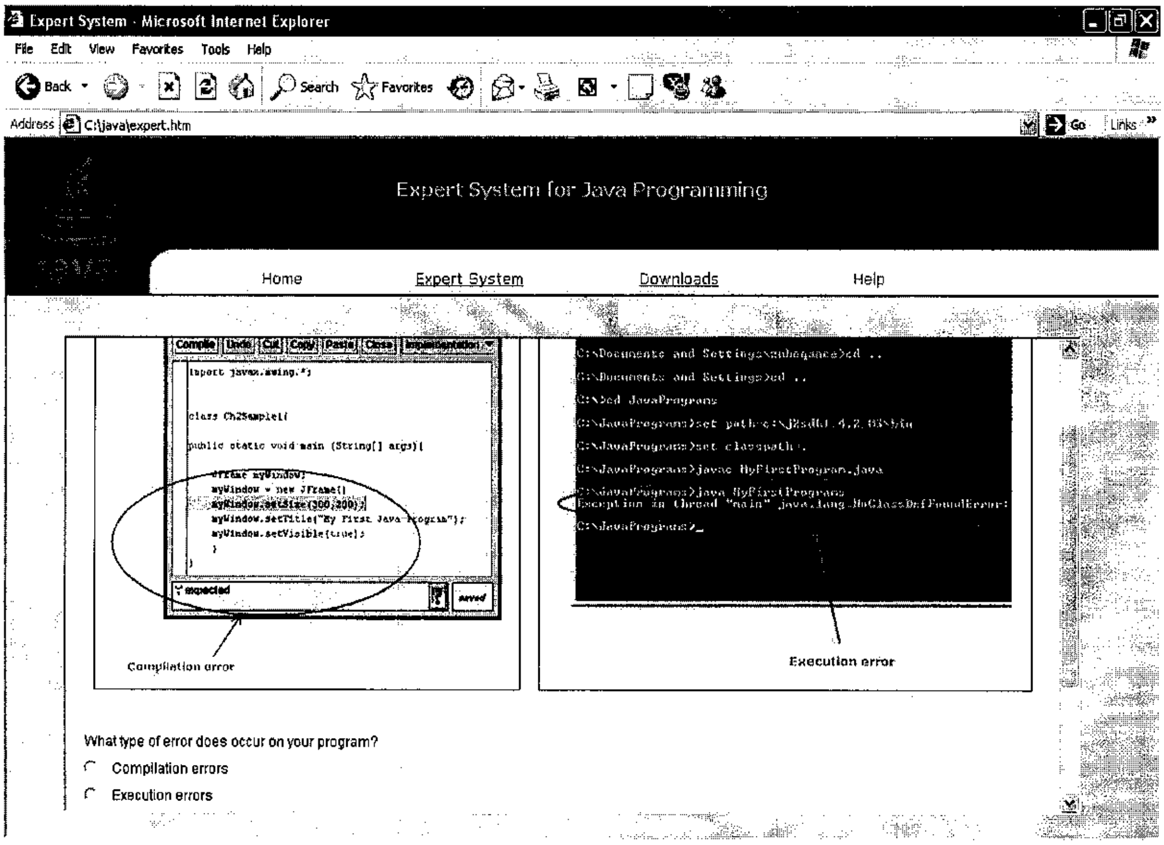
○ Not sure on how to run Java programs.

○ Compilation / Execution Errors

○ Not sure whether the applications (BlueJ and Java 2 SDK) have completely installed.

Figure 4.2: Initial Question of Java Programming Expert System

If user for example chooses the option 'Compilation / Execution Error' and click 'OK' button, then another question that requires user input appears. User has to identify the type of error occur on their Java program. See Figure 4.3 on next page.


Expert System - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back Search Favorites
Address C:\java\expert.htm
Go Links
Expert System for Java Programming
Home
Expert System
Downloads
Help
Compile Undo Cut Copy Paste Close Implementation
saved
Compilation error
Execution error

```
import javax.swing.*;


class Ch2Sample1{

public static void main (String[] args){

    JFrame myWindow;
    myWindow = new JFrame()
    myWindow.setSize(300,200);
    myWindow.setTitle("My First Java Program");
    myWindow.setVisible(true);
    }
}
```

```
; expected
```

```
C:\Documents and Settings\[illegible]>cd ..

C:\Documents and Settings>cd ..

C:\>cd JavaPrograms

C:\JavaPrograms>set path=c:\j2sdk1.4.2_03\bin

C:\JavaPrograms>set classpath=.

C:\JavaPrograms>javac MyFirstProgram.java

C:\JavaPrograms>java MyFirstProgram
Exception in thread "main" java.lang.NoClassDefFoundError:

C:\JavaPrograms>_
```

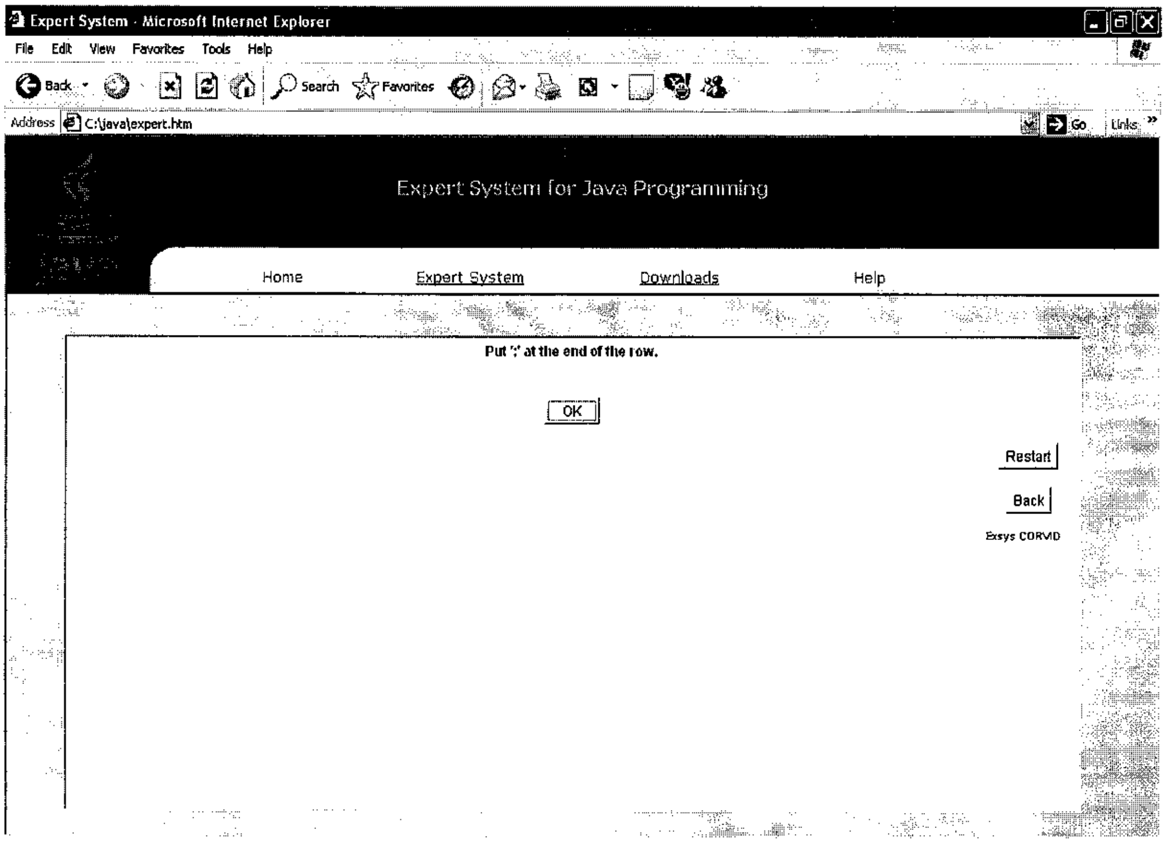
What type of error does occur on your program?

○ Compilation errors

○ Execution errors

Figure 4.3: Type of Error Does Occur

There are images to give clear view to the user to choose the right error they are attempting to. The user selects the answer based on what they are facing and the solution they desired to have. There might be some more next question may be asked to user and then will drive them to the solution and action need to be taken to correct the error in the Java program. An example of the final result or solution is shown on Figure 4.4. *(See next page)*



Expert System - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back
Search
Favorites
Address C:\java\expert.htm
Go
Links
Expert System for Java Programming
Home
Expert System
Downloads
Help

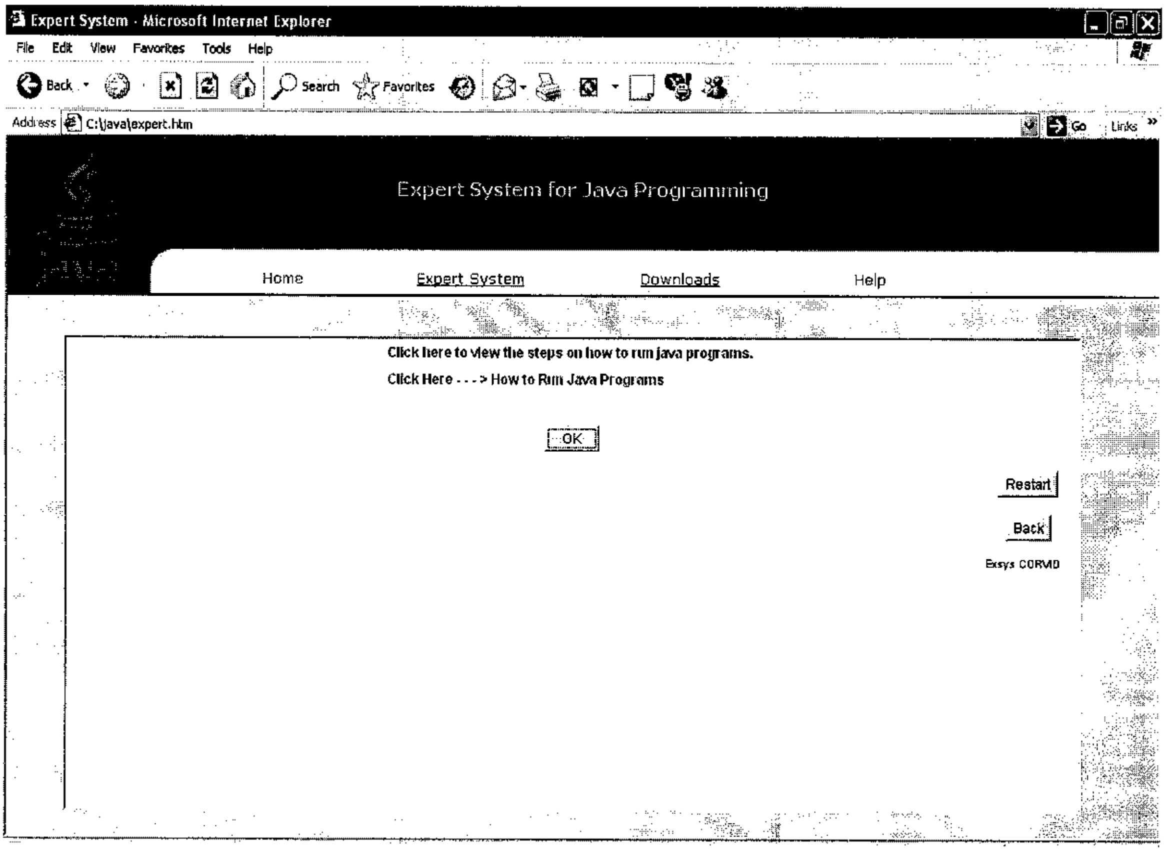
Put ';' at the end of the row.

OK
Restart
Back
Exsys CORVID

Figure 4.4: Final Result (Solution)

From the result shown above, user may take an action as being instructed to correct their Java programming code. User can click 'Restart' button to restart using the system or click the 'Back' button if they mistakenly click to the wrong option.

Since this project title is 'A Multimedia Enhanced Expert System for Java Programming', I have added some multimedia element to this system. If user choose the answer 'Not sure on how to run Java programs' for the initial question, this will lead them to this page.


Expert System - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back Search Favorites
Address C:\java\expert.htm
Go Links
Expert System for Java Programming
Home
Expert System
Downloads
Help

Click here to view the steps on how to run java programs.

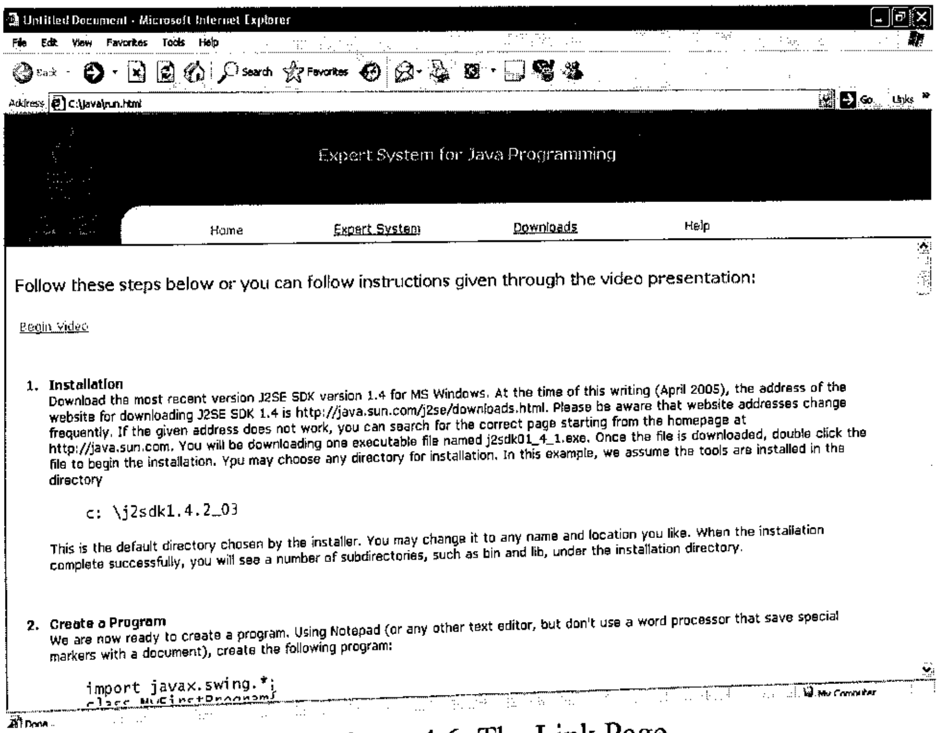
Click Here - - - > How to Run Java Programs

OK
Restart
Back
Exsys CORVID

Figure 4.5: Hyperlink Page

This page is linked to another page which contained the multimedia element. The hyperlink is highlighted when the user rollover the mouse onto the blue color font. User has to click at the hyperlink to go to the next page a new window will appear.

User have the option to read the instruction by them by reading it on the page shown below in Figure 4.6 or view and follow the instruction giving by the video presentation as shown in Figure 4.7.


Untitled Document - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back Search Favorites
Address C:\Java\run.html
Go Links
Expert System for Java Programming
Home
Expert System
Downloads
Help
Done
My Computer

Follow these steps below or you can follow instructions given through the video presentation:

Begin Video

1. **Installation**
   Download the most recent version J2SE SDK version 1.4 for MS Windows. At the time of this writing (April 2005), the address of the website for downloading J2SE SDK 1.4 is http://java.sun.com/j2se/downloads.html. Please be aware that website addresses change frequently. If the given address does not work, you can search for the correct page starting from the homepage at http://java.sun.com. You will be downloading one executable file named j2sdk01_4_1.exe. Once the file is downloaded, double click the file to begin the installation. Ypu may choose any directory for installation. In this example, we assume the tools are installed in the directory

   ```
   c: \j2sdk1.4.2_03
   ```

   This is the default directory chosen by the installer. You may change it to any name and location you like. When the installation complete successfully, you will see a number of subdirectories, such as bin and lib, under the installation directory.

2. **Create a Program**
   We are now ready to create a program. Using Notepad (or any other text editor, but don't use a word processor that save special markers with a document), create the following program:

   ```
   import javax.swing.*;
   ```

Figure 4.6: The Link Page

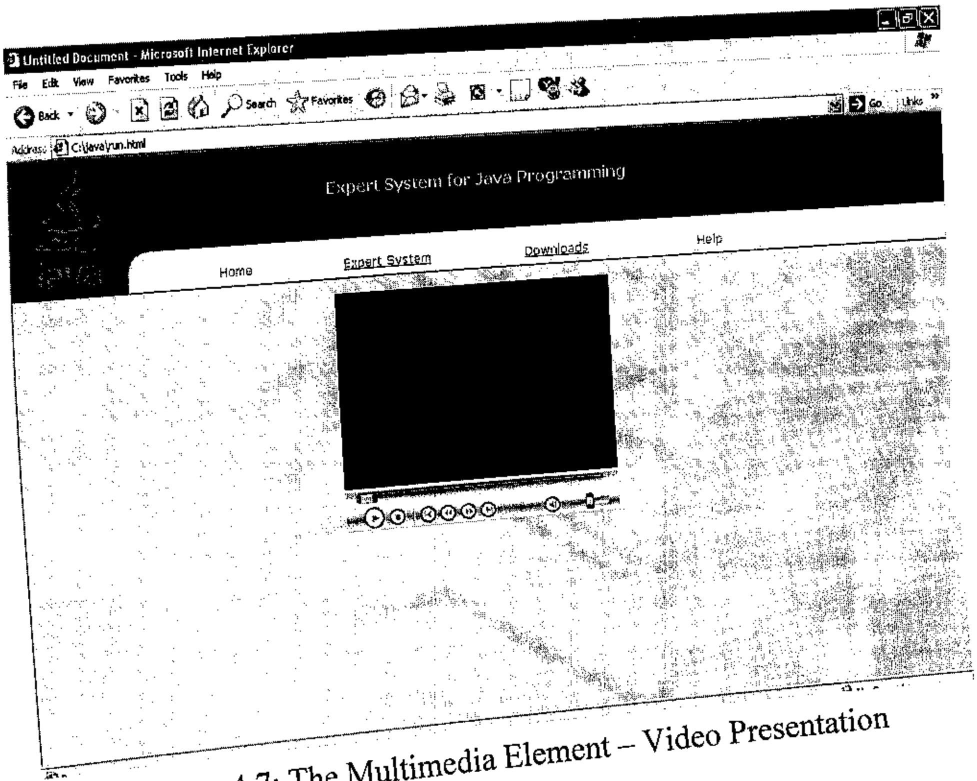

Untitled Document - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back Search Favorites
Address C:\Java\run.html
Go Links
Expert System for Java Programming
Home
Expert System
Downloads
Help

Figure 4.7: The Multimedia Element – Video Presentation

4.2.2 Test Cases

A set of prepared test cases (errors in Java programs) is executed on Java Programming Expert System. The results are based on the input respond by the user to the system. This approach is a black box approach, where inputs and outputs are significant.

Testing is also conducted in an exhaustive manner, where test cases cover possible combinations of input values. Here, the knowledge engineer ensures that all the input combination of Java program problems result in the correct solution, as have been obtained from knowledge acquisition. By doing this, it assures logical completeness and consistency of this system.

Test cases also provided by students, tutors and lecturers themselves. All these groups of users look at different aspect of the expert system. Test cases from all these different sources allow the performance of Java Programming Expert System to be verified using the best combination and most comprehensive test. This coverage of sources in such a testing plan allows this system to meet its objectives, which is to help and guide user to solve problem encounter while doing Java programming. Comprehensive testing with different user groups further enhances Java Programming Expert System's credibility and quality.

4.3 Findings

4.3.1 Advantages of Expert Systems

The first and the most powerful advantages of expert system is in term of permanence, that expert systems has the extra ability which its does not forget, but human experts may. The second advantage is in term of reproducibility. Many copies of an expert system can be made, but training new human experts is time-consuming and expensive.

In term of efficiency, expert system can increase throughput and decrease personnel costs. Although expert systems are expensive to build and maintain, they are inexpensive to operate. Development and maintenance costs can be spread over many users. The overall the cost can be quiet reasonable when compared to expensive scarce human experts.

Expert system can maintain the consistency. With expert systems, similar transactions handled in the same way. The system will make comparable recommendations for like situations. While human are by decency effects (most recent information having a disproportionate impact on judgment) primacy effects (early information dominates the judgment).

An expert system can provide permanent documentation of the decision process, review all the transactions, while a human expert can only review a sample. Since expert systems require knowledge acquisition where as the knowledge engineer has to gain knowledge from multiple experts, the knowledge of multiple human experts can be combined to give a system more breadth than a single person is likely to achieve.

Other advantages of expert systems are fraud or errors can be prevented, information is available sooner for decision making, reduce the risk of doing business, and consistency in decision making.

4.3.2 Disadvantages of Rule-Based Expert Systems

Some of the disadvantages of experts system are discussed here. In addition to a great deal of technical knowledge, human experts have common sense which it is not yet known how to give expert system common sense. Human experts can respond creatively to unusual situations but expert systems cannot. Human expert automatically adapt to changing environments, but expert system must be explicitly updated. Case-based reasoning and neural networks are methods that can incorporate learning.

Other disadvantages of expert systems are expert system are currently dependant on symbolic input and are not good at recognizing when no answer exists or when the problem is outside their area of expertise.

4.3.3 Criticisms of Expert Systems

When the rule set for an expert system is written, the knowledge of humans are observed. Video tapes, interviews, protocol, and other techniques are used to try to capture the thought process of experts. A problem with expert system is writing the rules themselves. Thought processes that are highly rule-oriented are easier to write than ones that rely more on creativity or intuition. Another problem is that often experts themselves disagree. Different experts might take different courses of action or go through different thought processes when give the same problem to solve. Thus there is disagreement in the professional community about the validity of expert systems.

4.4 Discussion

Expert system captures and deliver knowledge that is not easily represented using traditional computing approaches. They can retain the knowledge and experience of anyone in an organization (including people who are retiring), pre-process information to increase an expert's productivity, or allow someone with less training to perform functions at a higher level. Expert systems can be used to gain access to expertise immediately, around the clock, by many people at the same time.

An expert system is composed of two independent parts: an inference engine and a knowledge base. The inference engine is the control structure of the program that implements the knowledge represented in the knowledge base. The knowledge base is where the real power of the expert system resides: the coded pool of rules, insights and knowledge that the person doing that task brings to bear on it. This two part structure results in two important features of expert systems:

- It allows the system to be modified, updated and expanded more readily than traditional programs making it easier to keep the system current with changes in the field, or with changes in users' requirements, and

- It allows the system to provide an explanation of the reasoning behind it's conclusions which is necessary to provide the credibility and confidence that people require before routinely accepting it's advice.

CHAPTER 5
CONCLUSION AND RECOMMENDATION

A Multimedia-Enhanced Expert System for Java Programming will be one of the sources that can be referred by students in learning Java programming language. In developing the expert system for Java programming, I have understood the underlying concept of AI and expert system. To gather some information needed I have analyzed and done observations on common problems occur while using Java programming language. I have examined the rule based expert system approach in constructing a computer based knowledge education for Java programming and applied the theoretical knowledge in developing a good and successful expert system for Java programming.

After 14 weeks doing this project, I managed to complete this project and come up with the Multimedia Enhanced Expert System for Java Programming. I have done research and observations to gather all the data needed to make this project successful. All the work done has the relevancy and the objectives specified earlier has been achieved.

Regarding future work for expansion and continuation of this project, I suggest to have many more references to acquire the knowledge which means involves many human experts. This is to get more data and create more rules into the expert system. This system will be updated time to time since Java programming language is frequently emerge and come up with a new trend.

REFERENCES

[1] Elliot B. Koffman & Ursula Wolz, *Problem Solving with Java*. 2002. Second Edition

[2] Herbert Schildt, *Java 2 A Beginner's Guide*. 2003. Second Edition

[3] Kenneth A. Lambert & Martin Osborne, *Java A Framework for Programming and Problem Solving*. 2002. Second Edition

[4] Edwards, Alex & Connell, *N.A.D. Expert Systems in Accounting*. 1989. fordshire,UK: Pentice Hall

[5] Jaime Nino & Frederick A. Hosch, *An Introduction to Programming and Object oriented Design using Java.* 2002. First Edition

[6] Rich, Elaine & Knight. *Artificial Intelligence*. Second Edition. 1991, New York

[7] C. Thomas Wu, *An Introduction to Object-Oriented Programming with Java. 2004*. Third Edition

[8] http://www.comp.lancs.ac.uk/~kristof/research/notes/clipsvsjess/

[9] http://rds.yahoo.com/S=2766679/K=Jess+expert+system+shell/

[10] http%3A//www.schoultz.nu/archives/000302.html

[11] http://www.iceis.org/iceis2001/workshops/jess/Jess-leaflet.pdf

[12] http://en.wikipedia.org/wiki/Expert_system

[13] http://builder.com.com/5100-6401-1046208.html

[14] Exsys CORVID Manual Book

[15] http://www.ajrhem.com/EXPERT.pdf

APPENDICES

APPENDIX I : Questionnaire

UNIVERSITI TEKNOLOGI PETRONAS

Final Year Project
A Multimedia-Enhanced Expert System for Java Programming

Questionnaire – Problems in Java Programming

This questionnaire is developed on a purpose of finding information regarding to Java programming problems faced by the UTP students. The information will then be used in the development of a final year project titled A Multimedia Enhanced Expert, System for Java Programming. This questionnaire is anonymous and confidential. Your co-operation is very much appreciated.
(*Note: You may tick more than one answer*)

1. What is the most inefficient value you found in the 'BlueJ' Application?
   □ Interface
   Define interface: ....
   (Examples: toolbox, compiler, window form, etc)
   □ Programming
   □ Others: ....

2. What is the most inefficient value you found in the Java 2 SDK (Software Development Kit)?
   □ Window
   Define window: ....
   □ The commands use
   □ Others: ....

3. What kind of error you usually get?
   □ Syntax errors
   □ Execution errors
   □ Others: ....

4. Did you manage to overcome the error problem?
   □ Yes
   □ Most of the time
   □ Sometimes
   □ No

5. Does the compiler helpful to you?
   □ Yes
   □ Most of the time
   □ Sometimes
   □ No

6. How do you solve the problem?
   □ Find out by yourself
   □ Ask lecturer
   □ Ask someone else
   □ Others: ....

Give your comments regarding the application (BlueJ and j2sdk1.4.2):

....

....

....

APPENDIX II : GANTT CHART

No	Detail / Week	1	2	3	4	5	6	7	8	9	10	11	12	13	14
1	Topic Confirmation	▨													
	Requirement Definition	▨													
	• Introduction	▨													
	• Objective	▨													
	• Literature Review / references	▨													
2	System and Software Design		▨	▨											
	• Project Planning		▨	▨											
	• Choose Tools		▨	▨											
	• Schematic Diagram		▨	▨											
	• Develop Expert System Rules		▨	▨											
3	Submission of Preliminary Report			■											
4	Develop Prototype			▨	▨	▨	▨	▨	▨						
	• Convert Rules			▨	▨	▨	▨	▨	▨						
	• Design System Interface			▨	▨	▨	▨	▨	▨						
	• Review with Supervisor			▨	▨	▨	▨	▨	▨						
5	Submission of Progress Report									■					
6	Enhance Prototype									▨	▨	▨	▨	▨	
	• Pre EDX											▨			
7	Submission of Dissertation Final												■		
8	Oral Presentation													■	
9	Submission of Project Dissertation														■

▨ Development phases

■ Dateline for report and presentations

APPENDIX III : HTML Code (System Interface)

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN">
<html><head>
<title>Untitled Document</title>
<meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1">
</head>
<body bgcolor="#6699CC" background="image/banner3.jpg">
<div align="center"><br>
 <strong><font color="#CCCCCC" face="Verdana">Expert System for Java
Programming<br> <br> <br> <br>
 </font></strong>
 <table width="671" border="0" cellpadding="0" cellspacing="5">
  <tr>
   <td width="171"><div align="center"><font face="Verdana"><font size="2"><font
color="#0099FF"><font color="#0066FF"><a href="index1.htm"
target="mainFrame">Home</a></font></font></font></font></div></td>
   <td width="192"><div align="center"><font color="#0066FF" size="2"
face="Verdana"><a href="ExpertSystem.html" target="mainFrame">Expert
     System</a></font></div></td>
   <td width="171"><div align="center"><font face="Verdana"><font size="2"><font
color="#0099FF"><font color="#0066FF"><a href="install.html"
target="mainFrame">Downloads</a></font></font></font></font></div></td>
   <td width="171"><div align="center"><font face="Verdana"><font size="2"><font
color="#0099FF"><font color="#0066FF"><a href="credit.html"
target="mainFrame">Credit</a></font></font></font></font></div></td>
   </tr>
 </table>
 <strong><font color="#CCCCCC" face="Verdana"> </font></strong></div>
</body>
</html>
```

```
<HTML>
<HEAD>
<META HTTP-EQUIV="Content-Type" CONTENT="text/html; charset=windows-
1252">
<TITLE>Expert System</TITLE>
</HEAD>
<BODY BGCOLOR="#FFFFFF" background="image/banner.jpg" TEXT="#990000"
LINK="#0000FF">
<DIV ALIGN="right">
 <div align="center"><br>
  <applet
codebase = "./"
code = "Corvid.Runtime.class"
name = "CorvidRuntime"
archive = "ExsysCorvid.jar"
width = 900
height = 410
hspace = 0
vspace = 0
align = middle >
   <param name = "KBBASE" value = "" >
   <param name = "KBNAME" value = "system.cvR">
   <param name = "KBWIDTH" value = "700">
   The expert system would be running here but your browser has Java Applets
   disabled or does not support Java Applets.
  </applet>
 </div>
 <DIV ALIGN="right"></DIV>
</DIV>
</BODY>
</HTML>
```

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN">
<html><head>
<title>Untitled Document</title>
<meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1">
</head>
<body background="image/banner.jpg"><div align="center">
<embed src="video/run.avi" autostart="false" width="559" height="443"/></div>
</body>
</html>
```

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN">
<html><head>
<title>Untitled Document</title>
<meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1">
</head>
<body background="image/banner.jpg">
<table width="767" border="0" cellpadding="0" cellspacing="5">
  <tr>
    <td width="57">&nbsp;</td>
    <td width="695"><p>&nbsp;</p>
      <p><font color="#0066FF" size="2" face="Verdana">You can download Java
installers available below:</font></p>
      <ul> <li><font color="#0066FF" size="2" face="Verdana"><a
href="installer/bluej.exe">BlueJ</a></font></li>
        <li><font color="#0066FF" size="2" face="Verdana"><a
href="installer/j2sdk1_4_2_03windowsi586p.exe">J2SDK1.4.2_03
          </a></font></li>
      </ul></td>  </tr>
</table>
</body>
</html>
```

```html
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN">
<html>
<head>
<title>How to Run Java Programs</title>
<meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1">
</head>
<body background="image/banner.jpg">
<p align="right"><font color="#FF9933" size="2" face="Verdana"><strong><a
href="ExpertSystem.html"><br>
 </a></strong></font></p>
<p><font color="#0033FF" size="3" face="Verdana"><strong>Follow these steps below
 or you can follow instructions given through the video presentation:
</strong></font></p>
<table width="948" border="0" cellpadding="0" cellspacing="5">
 <tr valign="top">
  <td colspan="2"><div align="center">
    <p align="left"> <a href="embed.html"><font color="#0099FF" size="2"
face="Verdana">Begin
     Video </font></a></p>
    <p align="left">&nbsp;</p>
    </div></td>
 </tr>
 <tr valign="top">
  <td width="29"><div align="center"><font size="2"
face="Verdana"><strong>1.</strong></font></div></td>
  <td width="904"><p><font size="2" face="Verdana"><strong>Installation<br>
    </strong>Download the most recent version J2SE SDK version 1.4 for MS
    Windows. At the time of this writing (April 2005), the address of the
    website for downloading J2SE SDK 1.4 is http://java.sun.com/j2se/downloads.html.
    Please be aware that website addresses change frequently. If the given
    address does not work, you can search for the correct page starting from
    the homepage at http://java.sun.com. You will be downloading one executable
```

```
      file named j2sdk01_4_1.exe. Once the file is downloaded, double click
      the file to begin the installation. Ypu may choose any directory for installation.
      In this example, we assume the tools are installed in the directory</font></p>
     <p><font face="Arial">
     <blockquote><font face="Lucida Console">c: \j2sdk1.4.2_03</font></blockquote>
     </font></p> <p><font size="2" face="Verdana">This is the default directory
      chosen by the installer. You may change it to any name and location you
      like. When the installation complete successfully, you will see a number
      of subdirectories, such as bin and lib, under the installation directory.
      </font></p>
     <p>&nbsp;</p></td>
   </tr>
   <tr valign="top">
    <td><div align="center"><font size="2"
 face="Verdana"><strong>2.</strong></font></div></td>
    <td><p><font size="2" face="Verdana"><strong>Create a Program<br>
      </strong>We are now ready to create a program. Using Notepad (or any other
      text editor, but don't use a word processor that save special markers
      with a document), create the following program:</font></p>
     <p><font face="Lucida Console">
     <blockquote>
      <p>import javax.swing.*;<br>
       class MyFirstProgram{<br>
       <font color="#FFFFFF">____</font>public static void main (String[] arg)
       {<br>
       <font color="#FFFFFF">________</font>JOptionPane.ShowMessageDialog(null,
       &quot;It works!&quot;);<br>
       <font color="#FFFFFF">________</font>System.exit(0);<br>
       <font color="#FFFFFF">____</font>}<br>
       } </p>
     </blockquote>
     </font> <p><font size="2" face="Verdana">Type in the program exactly as
    shown, making sure the uppercase and lowercase letters are entered correctly.
    Note: Asimple text editor, such as Notepad, will display the code in black
    only.</font> </p>
   <p>&nbsp; </p></td>
 </tr>
 <tr valign="top">
  <td><div align="center"><strong><font face="Verdana"><font
size="2">3.</font></font></strong></div></td>
  <td><p><strong><font size="2" face="Verdana">Save the Program<br>
    </font></strong><font size="2" face="Verdana">Before we complie and run
    the program, let's save the program. First create a folder. <em>(Note:
    We use the words folder and directory synonymously.) </em>For this example,
    we will create a folder named <font face="Lucida Console">JavaPrograms</font>
    under the C:drive.</font></p>
   <p><font size="2" face="Lucida Console">
   <blockquote>
    <p>c:\JavaPrograms</p>
   </blockquote>
   </font> <p><font size="2" face="Verdana"></p>
   <p>Save the program by selecting the menu choice <font face="Lucida
Console">File/Save</font>
    of Notepad and giving the name <font face="Lucida
Console">MyFirstProgram.java</font>.
    Put this file in the <font face="Lucida Console">C:\JavaPrograms</font>
    folder. The name of the class is <font face="Lucida
Console">MyFirstProgram</font>,
    so we save it as <font face="Lucida Console">MyFirstProgram.java</font>.
    If you name the program <font face="Lucida Console">XYZ</font>, then save
    it as <font face="Lucida Console">XYZ.java</font>. Note that it is case-sensitive.
    When you use Notepad, be careful that the file is not saved as <font face="Lucida
Console">MyFirstProgram.txt</font>.
    make sure there's no <font face="Lucida Console">txt</font> suffix appended
    to the filename. To avoid the automatic appending of the <font face="Lucida
Console">txt</font>
    suffix, don't forget to set the value for <font face="Lucida Console">Save
    as type</font> to <font face="Lucida Console">All Files</font>. </font></p></p>
   <table width="863" height="467" border="0" cellpadding="0" cellspacing="5">
    <tr>
     <td width="125">&nbsp;</td>
     <td width="568" background="image/untitled.bmp">&nbsp;</td>
     <td width="150">&nbsp;</td>
    </tr>
   </table>
   <p>&nbsp;</p></td>
 </tr>
 <tr valign="top">
  <td><div align="center"><strong><font face="Verdana"><font size="2">4.
</font></font></strong></div></td>
  <td><p><strong><font size="2" face="Verdana">&nbsp;Open a Command Prompt
Window<br>
    </font></strong><font size="2" face="Verdana">After the source file is
    created and saved properly, we are ready to compile and run it. We use
    a command prompt window to enter the commands for compiling and running
    Java programs. Open a command prompt window by selecting the <font
face="Lucida Console">Start/Run...</font>
    option and entering the text <font face="Lucida Console">cmd</font> in
    the text field of the <font face="Lucida Console">Run</font> dialog
box.</font></p>
   <table width="883" height="202" border="0" cellpadding="0" cellspacing="5">
    <tr>
     <td width="247" height="72">&nbsp;</td>
     <td width="356" background="image/cmd.bmp">&nbsp;</td>
     <td width="260">&nbsp;</td>
    </tr>
    </table>
    <p><font size="2" face="Verdana">Click the OK button. Acommand prompt
window
      appears on the screen:</font></p>
    <table width="890" height="351" border="0" cellpadding="0" cellspacing="5">
      <tr>
        <td width="111">&nbsp;</td>
        <td width="664" background="image/black.bmp">&nbsp;</td>
        <td width="95">&nbsp;</td>
      </tr>
    </table>
    <p><font size="2" face="Verdana">From this point on, all commands are entered
      in this window.</font></p>
    <p>&nbsp;</p></td>
  </tr>
  <tr valign="top">
   <td><div align="center"><strong><font face="Verdana"><font
size="2">5.</font></font></strong></div></td>
   <td><p><strong><font size="2" face="Verdana">Set the Environment<br>
      </font></strong><font size="2" face="Verdana">Before we can actually compile
      and run the program, we must set the environment. First change to the
      JavaPrograms directrory where the source file is stored by entering the
      command cd JavaPrograms (and preseeing the Enter key);</font></p>
    <table width="906" height="357" border="0" cellpadding="0" cellspacing="5">
      <tr>
        <td width="25">&nbsp;</td>
        <td width="754" background="image/set.bmp">&nbsp;</td>
        <td width="107">&nbsp;</td>
      </tr>
    </table>
    <p><font size="2" face="Verdana">Then enter the following two commands in
      sequence to set the environment:</font></p>
    <p><font size="2" face="Lucida Console"> <blockquote>
      <p><font face="Lucida Console">set patch=c:\j2sdk1.4.2_03\bin<br>
        set classpath</font><font face="Lucida Console">=.</font></p>
    </blockquote>
    <font face="Lucida Console"></p></font> <table width="905" height="352"
border="0" cellpadding="0" cellspacing="5">
      <tr>
        <td width="26">&nbsp;</td>
        <td width="760" background="image/set2.bmp">&nbsp;</td>
        <td width="99">&nbsp;</td>
      </tr>
    </table>
    <p><font size="2" face="Verdana">Enter the commands exactly as shown. Do
      not indtroduce any spaces between the equals sysmbols (=), for example.
      The first command sets the PATH environment variable so we can refer to
      the executable files in the bin subdirectory of <font face="Lucida
Console">c:\j2sdk1.4.2_03</font>.
      The second command tells the Java compiler and interpreter where to find
      the source files. The period (.) indicates the current directory. You
      need to enter the two commands only once.</font> </p></td>
  </tr>
  <tr valign="top">
    <td><div align="center"><strong><font face="Verdana"><font
size="2">6.</font></font></strong></div></td>
    <td><p><strong><font size="2" face="Verdana">Compile the Program<br>
      </font></strong><font size="2" face="Verdana">Finally we are ready to
      compile the program. To compile a Java source file, use the <font face="Lucida
Console">javac</font>
      command followed by the filename of the source file. Enter the following
      command exactly, that is, in a case-sensitive manner:</font></p>
    <p><font size="2" face="Lucida Console">javac MyFirstProgram.java</font></p>
    <table width="901" height="354" border="0" cellpadding="0" cellspacing="5">
    <tr>
     <td width="43">&nbsp;</td>
     <td width="743" background="image/javac.bmp">&nbsp;</td>
     <td width="95">&nbsp;</td>
    </tr>
   </table>
   <p><font size="2" face="Verdana">After a moment of pause, when there's no
    error in the program, the prompt to enter the next commands appears. An
    error message will appear if there's in an error. If that happens, go
    back to Notepad and check the program. Make any necessary changes and
    save it again. Then enter the <font face="Lucida Console">javac</font>
    command again. </font></p>
   <p><font size="2" face="Lucida Console"><br>
    </font></p></td>
 </tr>
 <tr valign="top">
  <td><div align="center"><strong><font face="Verdana"><font
size="2">7.</font></font></strong></div></td>
  <td><p><strong><font size="2" face="Verdana">Run the Program<br>
    </font></strong><font size="2" face="Verdana">After the successful compilation
    of the program, we are finally ready to run the program by executing its
    bytecode file. To run the program, we use the <font face="Lucida
Console">java</font>
    command followed by the name of the bytecode file (with no suffix). Enter
    the command</font></p>
   <p><font size="2" face="Verdana">
   <blockquote>java MyFirstProgram</blockquote>
   </font>
   <p><font size="2" face="Verdana"></p> and press the Enter key. the program
    starts and a message dialog appears on the screen:</font></p>
   <table width="898" height="355" border="0" cellpadding="0" cellspacing="5">
    <tr>
```

```
     <td width="80">&nbsp;</td>
     <td width="711" background="image/java.bmp">&nbsp;</td>
     <td width="87">&nbsp;</td>
    </tr>
   </table>
   <p><font size="2" face="Verdana">Close this message dialog by clicking its
    OK button. The programs termintaes and another prompt appears on the command
    prompt window:</font></p>
   <table width="894" height="354" border="0" cellpadding="0" cellspacing="5">
    <tr>
     <td width="66">&nbsp;</td>
     <td width="728" background="image/congrate.bmp">&nbsp;</td>
     <td width="80">&nbsp;</td>
    </tr>
   </table>
   <p><font size="2" face="Verdana">Congratulations! You Have successfully
    executed your first Java program.</font></p></td>
 </tr>
 <tr valign="top">
  <td><div align="center"><font face="Verdana"><font
size="2"></font></font></div></td>
  <td><font size="2" face="Verdana">&nbsp;</font></td>
 </tr>
</table>
<p><font color="#0033FF" size="2" face="Verdana"></font> </p>
</body>
</html>
```

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN">
<html>
<head>
<title>Untitled Document</title>
<meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1">
</head>
<body background="image/banner.jpg">
<table width="915" border="0" cellpadding="0" cellspacing="5">
  <tr>
    <td width="57">&nbsp;</td>
    <td width="843"><p align="center">&nbsp;</p>
      <p align="center">&nbsp;</p>
      <p align="center">&nbsp;</p>
      <p align="center"><font color="#0066FF" size="2" face="Verdana"><strong><font
size="3">Developer:</font></strong></font></p>
      <p align="center"><font color="#0066FF" size="3" face="Verdana"><strong>Siti
        Zubaidah bt. Jerai</strong></font></p>
      <p align="center"><strong><font color="#0066FF" size="3"
face="Verdana">2751</font></strong></p>
      <p align="center"><strong><font color="#0066FF" size="3"
face="Verdana">Information
        Technology</font> </strong></p>
      <p align="center"><strong><font color="#0066FF" size="3"
face="Verdana">Universiti
        Teknologi Petronas</font></strong></p>
      </td>
  </tr>
</table>
<p>&nbsp;</p>
</body>
</html>
```