**GROUP ASSIGNMENT 2**

Expert Systems with Uncertainty

**ARTIFICIAL INTELLIGENCE**

CSC 3206

**TUESDAY P3 1**

Muhammad Awad Luckhoo (15042278)

Choong Kai Wern (15053648)

Teh Cuok Syen (15061567)

Ong Li Shen (15071863)

Mu Chun Khang (13079272)

Mah Qi Hao (16080111)

# TABLE OF CONTENTS

[TABLE OF CONTENTS 2](#_Toc485898682)

[QUESTION 1 3](#_Toc485898683)

[Literature Review 3](#_Toc485898684)

[Comparative Analysis 3](#_Toc485898685)

[Justification 3](#_Toc485898686)

[Expert System Design 3](#_Toc485898687)

[QUESTION 2 4](#_Toc485898688)

[Literature Review 4](#_Toc485898689)

[Quality of Recommendations 4](#_Toc485898690)

[QUESTION 3 5](#_Toc485898691)

[Contributions of Members 5](#_Toc485898692)

[Gantt Chart 5](#_Toc485898693)

[REFERENCES 6](#_Toc485898694)

# QUESTION 1

## Literature Review

Literature review about expert system

* What is expert system
* How it works (how the components interact)
* Applications of expert system
* Downfall

## Comparative Analysis

For the purpose of building an expert system, the paper discusses two main development tools: expert system shell and expert system programming language. To illustrate the differences between both expert system development tools, this section of the paper shall provide an exhaustive comparative analysis. In this context, an example is given for each respective expert system development tool. The tools are first discussed in terms of their common purpose. Subsequently, their distinct features are deliberated individually. This should provide readers with a good overall comparison of both expert system development tools.

It is imperative to note that expert systems use reason of knowledge to solve complicated problems. The systems are represented predominantly with if-then rules, and not the conventional procedural code. At this juncture, the development tools for expert systems must be specialized. An expert system development tool is simply a software development environment that contains the basic components of an expert system: knowledge base and reasoning engine. Typically, expert system development tools come with prescribe methods of building applications through configuration and instantiation of these components. Besides that, developers are also offered numerous choices when designing an expert system. The methodology, mode of knowledge representation, software development package and hardware to be implemented on are among the factors for consideration. Herein, all expert system development tools, be it expert system shell or expert system programming language, are crucial in aiding developers construct expert systems.

Expert System Shell

Expert system shells consist mainly of a user interface, inference engine and an editor to assist developers in building their knowledge base for expert systems. A great example of an expert system shell is Jess which stands for The Java Expert System Shell. Jess is an expert system shell written in Java entirely, driven by a Lisp-styled scripting language. Java provides the external mechanisms that generate and control the rules. To use Jess, the data must first be converted into text before it is handled by the interpreter.

Jess can be utilized in two ways, namely as a rule engine or as a general-purpose programming language. A rule engine is a special program that applies rules to large sets of data in an efficient way. A rule-based program can have up to thousands of rules, but Jess will have no problem applying them as data in the form of a knowledge base. In particular domains, the rules are represented as the heuristic knowledge of human experts, while evolving situations are represented by the knowledge base. On the other hand, as a general-purpose programming language, Jess can be extended easily as it directly accesses all Java classes and libraries. New commands can be written in Java or Jess to be integrated into Jess. Thus, Jess is very customizable when it comes to building applications.

The advantages of Jess include the ease of working with the code builder because it is an independent scripting language. Jess releases the burden from developers because they are not required to declare each rule as a set of nested class instances. On top of that, the programming effort for building the user interface and inference engine is also greatly reduced. Projects can be completed faster and cheaper in an efficient manner. However, Jess has the disadvantage of Java being disconnected from the rule engine. Normal Java syntax cannot be used to debug the syntax after external files and strings have been used to specify the rules. Besides that, Jess implements the Rete algorithm to match rules against the knowledge base. Since the algorithm trades space for speed, Jess requires relatively-large memory usage for moderately-sized programs.

Expert System Programming Language

Expert programming languages are high-level languages commonly used in expert systems due to their high capability in handling symbolic data efficiently. One example worth mentioning is Prolog, a language that is backward-chaining where it works backwards from a goal to find supporting facts. A Prolog program consists of a set of clauses whereby a clause is either a fact or a rule used to indicate a relationship between elements. Queries are entered using command-line tools provided by Prolog.

The process of deriving reason for a Prolog program can be broken down into multiple steps. Initially, a goal is given. Prolog searches the database from top to bottom for a fact that matches the goal. Then, a pointer is left where the match is found before Prolog instantiates the suitable variables. When a goal matches the head of a rule instead of a fact, the atoms within the rule’s body are treated as sub-goals that must all be satisfied to prove that the head is satisfied. Herein, the order of entries is important as the arrangement influences the number of search required to satisfy the goal and find a solution. In fact, there could be situations where Prolog may not find a solution even though a human can easily identify it from the given information.

Prolog has the advantage of being a powerful language that deduces the desired supplementary facts using strong built-in deduction system. The developer is freed from implementation details as he or she only needs to define what is required rather than indicate how it should be computed. Prolog also encourages modular programming and incremental developments, making program tracing and debugging a simple process. Nevertheless, it is very difficult to design a database that accurately represents relationships. Moreover, Prolog is not a suitable language to solve complex arithmetical computations.

## Justification

Justify Prolog

* Problem domain
* Knowledge domain

Prolog: Collections

In a very high level languages, other data structures other than list also available

Prolog: pattern matching

AI language usually have one critical property which is pattern matching because it will be needed when there is explicit structure decomposition that will also be used for associative retrieval and pattern action rules.

Prolog: Associative assertions

For a very high ES language, the facts about the symbol can be written as assertions. The set of assertions will have the associative data base that will allow access with more ease and natural. They can also used to deduce new facts.

PROLOG : non-determinism

Most important feature of high level ES language. Used to abstract points within unnecessary details and is one of the very few ways to cope with complex system. The existence quantifier of the logic non-determinism will permit the abstraction from the explicit disjuncts only one of which holds true, non-determinism control the permits an abstraction from a set of computation where only one of which need to succeed.

## Expert System Design

Describe and justify expert system design

* Approach / methodology
* System components
* Major design elements
* System strengths
* System weaknesses
* approach/methodology
  + This system will help the user to find out their interested subject or subject that suits them.
  + This system will filter out the subject that suitable for the user by asking them few preset questions.
  + What the user needs to do is answering the questions by choosing their answer from given choices.
  + A subject will only being suggested if all its pre-conditions or rules are achieved.
  + In order to prove a rule, the user should answer a question and the system will decide whether the user meets the rule.
* system components
  + questioning mechanism
  + answer parser
  + subject finder
  + degree finder
* major design elements
  + Agenda
    - Subject
    - degree
  + knowledge base
    - rules (subject rules, degree rules, logical thinking)
  + knowledge acquisition facility
    - question
    - answer
  + working memory
    - skills (science,logical thinking) after answering question
  + Inference engine
    - Backward chaining
  + Explanation facility
    - describe
* system strengths
  + Increased availability
    - Mass production of expertise since it can be made available on a computer
  + Reduced cost
    - Average cost of providing expertise is greatly lowered
  + Permanence
    - Expertise is permanent, lasts longer than the human expert.
  + Fast response
    - Sometimes real-time response is required, ES is more available than human expert
* weaknesses
  + no security
  + no error handling
  + does not include all the subjects available

# QUESTION 2

## Literature Review

Literature review about uncertainty in expert systems

* What is uncertainty
* Why should it be introduced in expert systems?
* Applications of expert systems with uncertainty

## Quality of Recommendations

Compare recommendations made by expert system without uncertainty and expert system with uncertainty

* Input
* Difference in output
* Justification
* Which one is better?

# QUESTION 3

## Contributions of Members

Muhammad Awad Luckhoo

Tasks here

Choong Kai Wern

Tasks here

Teh Cuok Syen

Tasks here

Ong Li Shen

Tasks here

Mu Chun Khang

Tasks here

Mah Qi Hao

Tasks here

## Gantt Chart

Gantt chart for planned and actual timeline

# REFERENCES

APA references here

[1] <http://www.jessrules.com/doc/61/intro.html>

[2] http://www.it.kmitl.ac.th/~pattarachai/ES/PDF4/Ch08-ESTools.pdf

[3] <http://developeriq.in/articles/2014/jul/14/expert-systems-programming-languages-tools-and-she/>

[4] <https://www.academia.edu/23352301/Tools_of_Development_of_Expert_Systems_A_comparative_study>

[6] <https://www.cs.unm.edu/~luger/ai-final2/JAVA/CH%2026_Case%20Studies%20-%20JESS%20and%20other%20Expert%20System%20Shells%20in%20Java.pdf>

[5] <http://www.iau.dtu.dk/teaching/31380/Jess/manual.pdf>

[7] http://http-server.carleton.ca/~aramirez/4406/Reviews/SRetchford.pdf