International Journal of Application or Innovation in Engineering & Management (IJAIEM)
Web Site: www.ijaiem.org Email: editor@ijaiem.org, editorijaiem@gmail.com
Volume 2, Issue 1, January 2013 ISSN 2319 - 4847

Student Counseling System: A Rule-Based Expert System based on Certainty Factor and Backward Chaining Approach

Prof. Tilotma Sharma[1], Prof. Navneet Tiwari[2], Prof. Khushboo Shah[3]

[1]BE(CS),MTech(IT), MITS Ujjain

[2]BE(CS),MCA, MITS Ujjain

[3] BE(IT),MTech(IT), MITS Ujjain

ABSTRACT

The proposed SCS expert system, aims to improve the method of selecting the best branch for student planning to take admission in engineering. The basic idea of the proposed work is to design an expert system which would test the student capabilities by asking questions of the particular branch and gives the result in the form of percentage based on certainty factor concept. The SCS is implemented as a rule-based expert system with two major components- knowledge base and inference engine. The developed inference engine is built using backward chaining method and having certainty factor for reasoning under uncertainty. CLIPS language is used as a tool for designing the SCS expert system.

Keywords: Expert System, rule-based system, backward chaining, certainty factor, CLIPS.

1. INTRODUCTION

Artificial Intelligence (AI) is the area of computer science focusing on creating machines that can engage on behaviors that humans consider intelligent. Expert systems (ES) are a branch of AI, and were developed by the AI community in the mid 1960s. An expert system can be defined as “an intelligent computer program that uses knowledge and inference procedures to solve problems that are difficult enough to require significant human expertise for their solutions [1].” From this definition it can be inferred that expertise can be transferred from a human to a computer and then stored in the computer in a suitable form that users can call upon the computer for specific advice as needed. Then the system can make inferences and arrive at a specific conclusion to give advices and explains, if necessary, the logic behind the advice.

An expert system has some main components, i.e., user interface, expert system database, knowledge acquisition facility, and inference mechanism. In addition, there is another component in some expert systems, i.e., explanation facility. User interface is software providing communication media between user and the system. The database consists of knowledge of expert level at certain subject. The knowledge may come from experts, journals, magazines, and other knowledge sources. Knowledge acquisition facility is software providing dialog facility between an expert and the system. This facility is used to input facts and rules based on knowledge development. Inference mechanism is software that is able to do reasoning using existing knowledge to produce a conclusion or final result. Explanation facility is useful in giving explanation to user why computer asks certain information from user and what ground used by the computer in concluding a condition. In order for the knowledge to be able to be used by the system, the knowledge should be represented in certain formats that are subsequently collected in a knowledge base. Expert system method in representing knowledge will influence system development, efficiency, and improvement [2].

A rule-based system is an expert system that contains a general rule-base and an inference engine. The inference engine retrieves rules from the rule-base to solve new problems based on the rules for similar problems stored in the rule-based system. In this way, a rule-based system can exhibit humanlike performance in that knowledge that can seemingly be acquired through experience [3][4].

The development of ES is implemented in CLIPS programming environment (C Language Integrated Production Sytsem) [5][6]. This programming tool is designed to facilitate the development of software to model human knowledge. CLIPS program is used by reason of the flexibility, the expandability and the low cost.

Certainty factor (CF) concept is used in SCS. CF have been quite popular among expert system developers since their appearance, because they are associated with a simple computational model that permits to estimate the confidence in conclusions being drawn.

The paper is organized as follows: section 2 describes the general idea of SCS expert system and problem identification, section 3 presents knowledge base representation and CLIPS rules, section 4 presents various testing performed , the result of the expert system is in section 5, and section 6 concludes the paper and outline direction for future research.

2. PROBLEM IDENTIFICATION

After completion of 12th class, those who are science students and opted mathematics, and wanted to take admission in engineering, then SCS expert system are designed for those students who can get guideline to take which branch in the engineering. The SCS expert system asks certain questions of particular branch, which student opted and depending on answers given by the student, the expert system gives percentage eligibility to take admission in that particular branch.

3. KNOWLEDGE REPRESENTATION

One of the well-know methods of representation of knowledge in the expert systems is the productive representation as the CLIPS (production system). SCS expert system is implemented in CLIPS using backward chaining algorithm. CLIPS keeps in memory a fact list, a rule list, and an agenda with activations of rules. Rules in CLIPS consist of patterns (usually related to facts) and actions. Actions are actually functions which typically have no return value and may perform insertion or deletion of facts to the fact list, printing of messages in an output device, etc. A rule is activated and put on the agenda, when all of its patterns match with facts in the fact list. When multiple activations are on the agenda, CLIPS automatically assigns to each of the activations a salience, according to which the corresponding rule will fire and thus its actions will be executed.

4. CERTAINTY FACTOR

In the mid 1980s, David McAllister, at MIT, developed a metric for `certainty factors' for use in an `expert system' (a type of computer program). A certainty factor is used to express how accurate, truthful, or reliable you judge a predicate to be. It is your judgment of how good your evidence is. The issue is how to combine various judgments.
An expert system should be able to work in uncertainty [1]. Theories are found to solve uncertainty such as classical probability, Bayesian probability, Hartley theory based on classical sets, Shannon theory based on probability, Dempster-Shafer theory, Zadeh's fuzzy theory and certainty facto. Certainty factor was introduced by Shortliffe Buchanan in designing MYCIN. Certainty factor (CF) is clinical parameter value given by MYCIN to show confidence level. Certainty factor is defined in equation 1 as follows [1]:

$$CF(H, E) = MB(H, E) - MD(H, E) \qquad (1)$$

CF(H,E) : certainty factor from hypothesis H influenced by evidence E. CF value is from –1 to 1. Value of –1 shows absolute uncertainty whereas value of 1 shows absolute certainty.
MB(H,E) : measure of increased belief to hypothesis H influenced by evidence E.
MD(H,E) : measure of increased disbelief on hypothesis H influenced by evidence E
McAllister defined the rule for adding two positive certainty factors like this:

$$CFcombine\ (CFa\ CFb) = CFa + CFb(1 - Cfa) \qquad (2)$$

The rule for adding two negative certainties is simple: Treat the two factors as positive and negate the result

$$CFcombine\ (CFe\ CFf) = -(CFcombine\ (-CFe\ -CFf)) \qquad (3)$$

The rule for adding positive and negative certainty factors is more complex:

$$CFcombine\ (CFg\ CFh) = (CFg + CFh) / (1 - \min\{|CFg|, |CFn|\}) \quad (4)$$

In diagnosing, an expert sometimes uses incomplete or uncertain data. In order for an expert system to be able to do reasoning like expert do, in spite of uncertainty and incomplete data, *Certainty Factor* can be used [7].

5. BACKWARD CHAINING

Backward chaining starts with a list of goals (or a hypothesis) and works backwards from the consequent to the antecedent to see if there is data available that will support any of these consequents. An inference engine using

backward chaining would search the inference rules until it finds one which has a consequent (**Then** clause) that matches a desired goal. If the antecedent (**If** clause) of that rule is not known to be true, then it is added to the list of goals (in order for one's goal to be confirmed one must also provide data that confirms this new rule). It is also called as **"Goal Driven Method"**. It is based on depth first search algorithm [8].
The standard backward-chaining control cycle is:
1. Check the conclusions of the rules to find all rules that can satisfy the top goal on the stack.
2. Process these rules one at a time:

a. Evaluate the conditions in the rules If part one at a time:

i. If the condition is currently unknown (that is, if there is not enough information currently known to determine whether the condition is satisfied) push a goal to make that condition known, and recursively invoke the system.

ii. If the condition is known to be unsatisfied, continue with the loop at Step2.

iii. If it was not possible to determine whether the condition was satisfied, continue with the loop at Step 2.

b. If all the conditions in the selected rule are satisfied, add to Working Memory the facts specified in the Then part of the rule, pop the goal off the stack, and return from this invocation of the system.

The system terminates with success when the goal stack is empty. It terminates with failure if the system runs out of rules to try in Step 2.
The time complexity of DFS is same of BFS i.e. $O(b^d)$. It is less demanding in space requirements, however, since only the path from the starting node to the current node needs to be stored. Therefore, if the depth cut off is d, the space complexity is just O(d).

BACKWARD CHAINING

1. Express associations between a goal (consequent) and the conditions it depends on (antecedents) as facts, and name this fact **domain-rule**.
2. The consequents of a domain-rule are the goals for be resolved.
3. If a domain-rule has an antecedent then each antecedent becomes a sub-goal.
4. Each sub goal is resolved by affirming or negated it based on the facts called **answer**.
5. If a sub-goal is affirmed then this antecedent is removed from the domain-rule's antecedents.
6. Repeat step 3 and 4 until there are no antecedents left.
7. When a domain-rule has no antecedent then fire the rule to create of a new fact called **conclusion**.
8. The conclusion consists of the goal along with certainty rating called "confidence-factor".
9. If two goals resolve to same conclusion then combine their certainties (confidence-factors) using the function **((100* (cf1 + cf2))-(cf1 * cf2))/100**

6. RESULTS

This paper presents a KBS for Student Counseling System. The SCS is implemented using CLIPS expert system language. During the test phase of system it never gave wrong output according to the rules used. CLIPS language does not support graphical interfaces, but can be integrated with other high level language like VC++. following are the snapshots of SCS expert system.

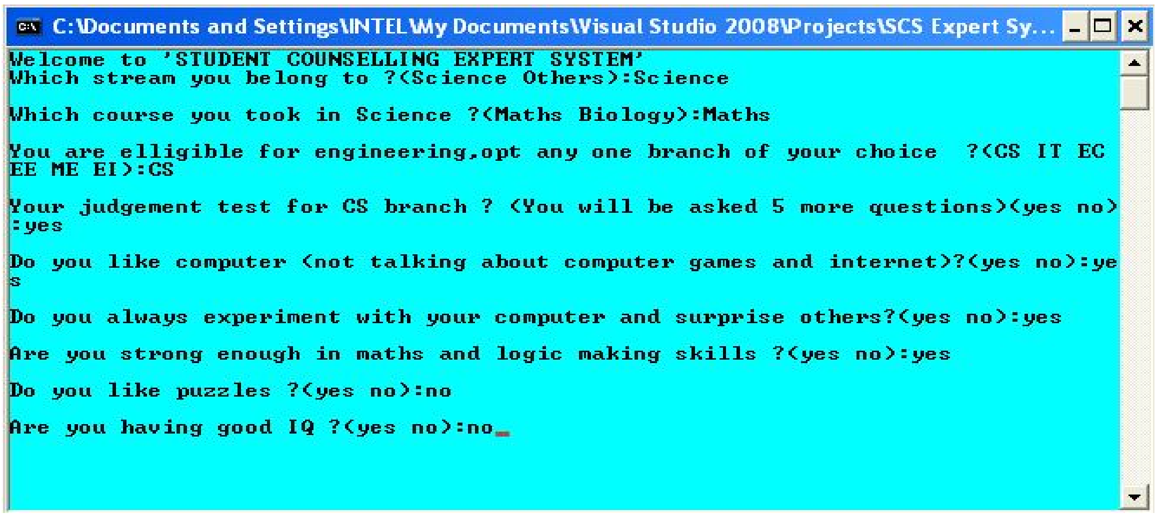

Fig1: User Input to the SCS for CS Branch

Fig1 shows how users interact with the SCS expert system. User enter the answer to the question asked by the expert system in the form of either yes or no.

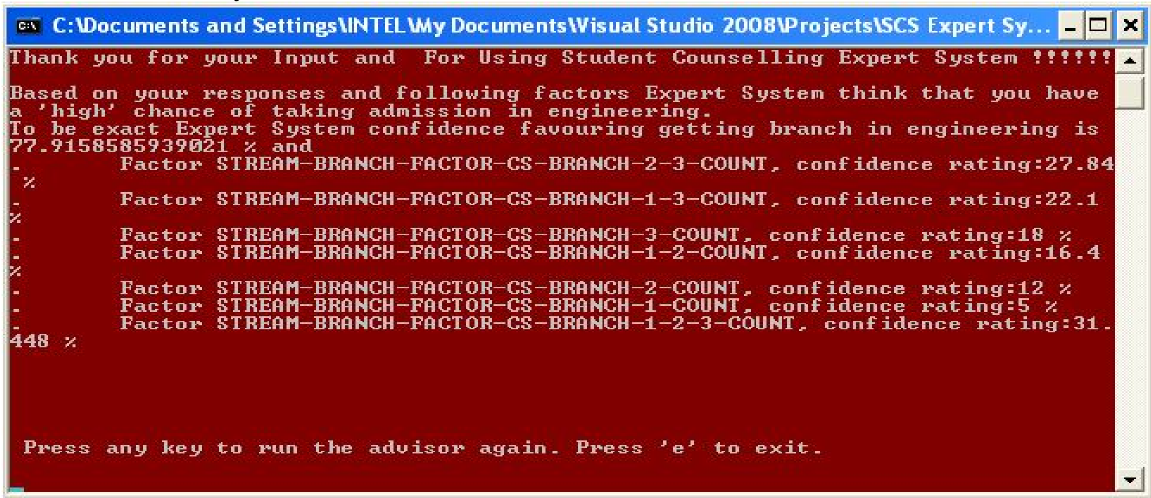

Fig2: Output of SCS ES for CS Branch

Fig 2 shows the output of the answering given by the expert system. Each question has been rated certain certainty factor by the expert of the respective field and based on answer given by the user the percentage eligibility to take a particular branch is given by the SCS ES.

7. CONCLUSION & FUTURE WORK

This paper presents the design and development of a proposed expert system to help the student to choose a particular branch of engineering according to the capability of answering the question asked by the expert system. The SCS expert system is a rule- based expert system and used CLIPS language to store knowledge base using backward chaining and certainty factor approach. Up till now CLIPS uses forward chaining but SCS expert uses backward chaining.

The SCS expert system is limited to only 6 branches of engineering i.e. CS, IT, EC, EE, EI, ME but can further be expanded to all other branches. Even this expert system system can be extended with every course to help student to select proper carrier.

REFERENCES

[1] Joseph Giarratano, Gary Riley (2004). *Expert Systems: Principles and Programming*, Fourth Edition.
[2] Turban, E., *Decision Support System and Expert Systems*, Prentice Hall International Inc., USA, 1995.
[3] Y.-Q. Zhang and others , Comeutational Web Intelligence (Series in Machine Perception and Artificial Intelligence (Vol. 58)),2004 Georgia State University, Atlanta, Georgia, USA .
[4] Engelbrecht A. P., 2002. Computational Intelligence An Introduction. John Wiley & Sons.
[5] Giarranto J. C., 1998. CLIPS User's Guide, Version 6.22, 1998.

[6] Jackson, P., 1999. Introduction to Expert Systems, Harlow, England: Addison Wesley Longman. Third Edition.

[7] Kusrini, 2007, Question Quantification to Obtain User Certainty Factor in Expert System Application for Disease Diagnosis, Proceedings of the International Conference on Electrical Engineering and Informatics, Indonesia.

[8] Knut Hinkelmann, *Forward Chaining Vs Backward Chaining*, University of Applied Sciences Northwestern Switzerland School of Business.