StuA: An Intelligent Student Assistant

Pooja Lodhi, Omji Mishra, Shikha Jain*, Vasvi Bajaj

Department of Computer Science, Jaypee Institute of Information Technology, Noida (India)

Received 7 September 2017 | Accepted 23 January 2018 | Published 16 February 2018

unir LA UNIVERSIDAD EN INTERNET

Abstract

With advanced innovation in digital technology, demand for virtual assistants is arising which can assist a person and at the same time, minimize the need for interaction with the human. Acknowledging the requirement, we propose an interactive and intelligent student assistant, StuA, which can help new-comer in a college who are hesitant in interacting with the seniors as they fear of being ragged. StuA is capable of answering all types of queries of a new-comer related to academics, examinations, library, hostel and extra curriculum activities. The model is designed using CLIPS which allows inferring using forward chaining. Nevertheless, a generalized algorithm for backward chaining for CLIPS is also implemented. Validation of the proposed model is presented in five steps which show that the model is complete and consistent with 99.16% accuracy of the knowledge model. Moreover, the backward chaining algorithm is found to be 100% accurate.

Keywords

Artificial Intelligence, Backward Chaining, CLIPS, Expert System, Rule-based System, Virtual Assistant.

DOI: 10.9781/ijimai.2018.02.008

I. Introduction

Artificial Intelligence is the science of making computers perceive their environment in a similar way as a human does and takes actions [1]. Artificial Intelligence is not a cognitive process rather it has an array of separate components of intelligence like learning, problem-solving, perception, and language understanding. Using artificial Intelligence, researchers are creating various systems that can mimic decision -making abilities of human based on past experiences and understand human thoughts. Such systems consist of a knowledge-base, inference engine, and user-interface. The knowledge-base has facts i.e., past experiences whereas inference engine has a set of rules that can infer a given situation while the user interface is used to interact with the user. Knowledge-base represents the knowledge that is stored by a model and is used to conclude new rules and check inconsistencies. The development of most of the virtual and intelligent software systems is implemented in CLIPS(C Language Integrated Production System, first released by NASA in 1988) [2-3] programming environment. CLIPS is a programming tool which is designed to ease the development of software that can model human knowledge. CLIPS program is used because it is flexible, expandable and has low cost. It is the only expert system shell for which it is claimed that the shell has been certified correct [27].

Virtual Assistants refer to software that can provide required information or perform work for human. There are mainly two types of virtual assistants: rule-based and stochastic. When there is lack of data, expert knowledge is used to design rule-based assistants also known as expert systems. While in presence of already tested large and valid information, stochastic virtual assistant can be designed using various machine learning techniques. A number of virtual assistants exist in different domains. A number of rule-based assistants are working in different domains such as counseling [4-7], prediction [8-10], diagnosis [11-19], design [20-21, 30], e-learning [28-29] and recommendations [31]. Some of the stochastic assistants are Chatbots, question answering systems and assistants like Apple's Siri, Google's Google Assistant, Amazon's Alexa, Microsoft's Cortana and Facebook's M.

All these virtual assistants are designed with an aim to replace a human being and assist the people around. These systems are doing really well in their domain. In the world of technology, people are more comfortable in communicating with a computer or any other electronic device than a human being. Sometimes people hesitate to interact and ask queries to a human being as either they want to keep it secret or they feel shy. Sometimes they want somebody's help but are not able to reach a right person to seek a help. This motivated us to design an interactive and intelligent student assistant situated in a close environment. It helps students who are new to the college environment. When a newcomer or a fresher enters in a college, he/she may not be familiar with the environment and seeks for the guidance from some experienced person like senior or teacher. But at the same time, some of them fear of being ragged or hesitate to ask. Hence, the new-comers remain unaware of the college environment. In such a scenario, the proposed student assistant, StuA, helps the newcomers and tries to provide the required information in a safe and sound manner to the students. It helps in familiarizing them with the rules and regulations of the college in a comfortable manner. StuA is equipped with the unique features which allow the user to ask questions on WHAT IS, WHERE, WHEN, WHAT HAPPEN format. Moreover, it facilitates real-time, low-cost expert-level assistance 24X7, unlike a human. The system based on its previously gained knowledge and beliefs is able to provide answers to most of the queries. The set of questions is not limited to only the prefixed questions. Nevertheless, to the best of our knowledge, no such virtual intelligent assistant exists till date. Moreover, it can be further customized to handle the queries of a new person in any environment such as a new office, organization, reception of hotels, hospitals, schools and malls.

StuA is a rule-based virtual assistant in the domain of education. The reason of designing it as rule-based is the lack of documented information. The rule-base system is designed with the help of experts

* Corresponding author.

E-mail addresses: shi_81@rediffmail.com

and general information in this domain. The implementation has been done in CLIPS. The main limitation of CLIPS is that it supports only forward chaining and does not allow backward chaining. As a result, many of the queries cannot be processed. Till date, no generic extension of CLIPS with backward chaining exists. A few researchers [22-23] have tried to add this functionality but for the specific domains only. So, here, we propose a generic model of backward chaining also, to extend the functionality of CLIPS so as to provide better inference mechanism. The whole model is designed using Java as the user interface. When java is integrated with CLIPS it provides the flexibility to redefine the output and with help of java, we can assert only relevant facts reducing the overhead of searching through irrelevant facts. It also aids the user by providing a friendly user-interface.So now, only the relevant rules are fired and the filtered and processed output is displayed.

The paper is organized as follows. In Section 2, the related work is discussed followed by the detail explanation of the proposed model (Section 3). Section 4 presents the implementation detail of the proposed model and Section 5 discusses the test results and validation of the proposed model. The Section 6 concludes the paper with limitations and provides direction for future work.

II. Related Works

Till date several systems are designed in various domains [4-21]. One of the expert systems in education domain is developed to help students in selecting the best branch who are planning to take admissions in engineering; this system is called Student Counseling System (SCS). Here they have used certainty factor to provide basis to their judgment. They have only chosen engineering branch as their domain for their counseling system. This expert system answers the queries in the form of 'yes' or 'no' only using forward chaining [4]. This system is rule-based system which uses CLIPS to answer the queries.

Another domain is diagnosing some of the eye diseases. It provides the expert guidance on eye diseases. The disadvantage of the system is that it is not taking the symptoms as input from the user. Rather, the expert system prompts menu and user has to select from it and, based on the disease, user has to answer queries in yes/no form [11]. Moreover, it is able to answer only the queries that can be derived either from the facts or from forward chaining. Various expert systems are being designed in the field of medical sciences. One of them is plant disease diagnosis system. They have used two methods to diagnose plant disease. One is Step by Step description and other is Graphical Representation System. The limitation of step by step description is the user is not allowed to enter his query, rather he has to select from a list of options. Hence search query is limited. Graphical Representation System provides more meaningful results when searched with keyword disease. Here the user has to only enter keywords. The limitation is he cannot enter the full description of disease [13]. The system proposed was the rule-based system which uses forward reasoning and pattern matching to answer the queries. Nevertheless, they concluded that expert systems is one of the successful methods which helped and supported users in making the right decisions in scenario where they have lack of knowledge.

Another expert system in the row deals with Diagnosis of Neuromuscular Disorders [15]. The proposed system is implemented using JESS. Here, the user is presented with a list of questionnaires about symptoms and possible treatments are suggested according to disorder. They designed different rules for different disorder on basis of knowledge acquired from experts. The limitation of this work is that it can detect only Cerebral Palsy, Multiple Sclerosis, Muscular Dystrophy and Parkinson's disease. Even other small diseases can't be detected using this system. Another research work deals with the work of diagnosis of rice plant disease. This expert system is also developed using JESS [16]. They have used SQL to store the data and extract information stored.

Similarly, there are different research works available in the field of system that assists people known as virtual assistants. Another example in domain of education is an expert system which aimed to assist Student in there major selection process. It is a prototype of an advisory expert system which can assist new and incoming students to select suitable majors they can apply to and the most convenient institutions they can attend. In this research work, user has to select from different courses and subjects and after confirmation final output is shown. They have designed a rule-based system in which they majorly defined three broad categories of rules to assist the students. The drawback of their work is that they have not tested their work on real-world cases; they have used fabricated test-cases to simulate their results [5]. A system PAS (Postgraduate Advisory System) is proposed which enables the students to select and get a plan for each semester without the need to consult advisors [6].The proposed system was different from other systems as it responded taking into account students thesis fields. The system is limited to one department; its rules are only defined for one department. Another expert system that is developed worked in Car Failure domain. Here, the author has categorized problems broadly into three categories. The drawback of the system is that user has to choose from a limited number of options [14]. He cannot input his own query.

TABLE I. Existing Rule-Based Systems

Domain	Paper	Category	Type	Criticism
Education	SCS [4]	Question Answering	Rule-Based	Only answer query in form of yes or no.
	Student Major Selection [5]	Assistant	Rule-Based	Application is not tested on real-world cases; fabricated test-cases used to simulate the results.
	PAS [6]	Assistant	Rule-Based	Limited to one department.
Medical Sciences	Eye Diseases Diagnosis [11]	Question Answering	Rule-Based	Symptoms are not taken as input from the user. User has to select from a list of symptoms designed by the expert. And it only answers query in form of yes or no.
	Plant Disease Diagnosis System [13]	Question Answering	Rule-Based	User is restricted to select from limited set of queries.
	Diagnosis Of Neuromuscular Disorders [15]	Question Answering	Rule-Based	This system only detects a limited number of disorders.
	Diagnosis Of Rice Plant Disease [16]	Question Answering	Rule-Based	-
Industry	Car Failure [14]	Assistant	Rule-Based	User is restricted to select from limited set of queries.

Moreover, there are virtual assistant such as Siri, Cortana, and Google assistant which are using some machine learning techniques to learn the procedure of doing the task. Cortana mostly answers general questions for which it pulls information from Bing. Whereas, Siri assists in all kind of works in a device like calling people, sending text messages, setting reminder, etc.

In conclusion, the virtual assistants can be designed in two ways: rule-based and machine learning based. For the domains, where data is not available in the form of corpus, it is difficult to apply some machine learning technique for decision making. In such cases, expert knowledge is required to design the systems (also known as expert systems). Most of the expert systems are using CLIPS and hence they are able to answer the queries using forward chaining only. They provide a fixed set of queries and user has to select one out of them only to get the answer which restricts the functionality of the system. Table I presents the summary of existing rule-based systems.

III. Proposed Model

Recognizing the need of a virtual assistant for the new-comers in a college, we propose an interactive and intelligent student assistant, StuA, which can help the new students in familiarizing them with the new environment, rules and regulations. It is a rule-based assistant because of lack of documented information. Furthermore, it tries to answer the queries of the students using inference mechanism without fixing the set of queries and hence providing a broader and unrestricted platform to resolve the doubts. The user is allowed to ask anything related to the domain and he need not pick the question from the already framed list. The domain knowledge is stored in the form of knowledge-base drilled by knowledge engineers. The proposed assistant responds to the question not only using forward chaining. Rather, the CLIPS tool is extended to empower the inference mechanism of the tool with backward chaining algorithm as well. The extended version of CLIPS with backward chaining is capable to handle most of the queries such as

Inference rule: Student break rule => student has to pay fine

Query: when student has to pay fine?

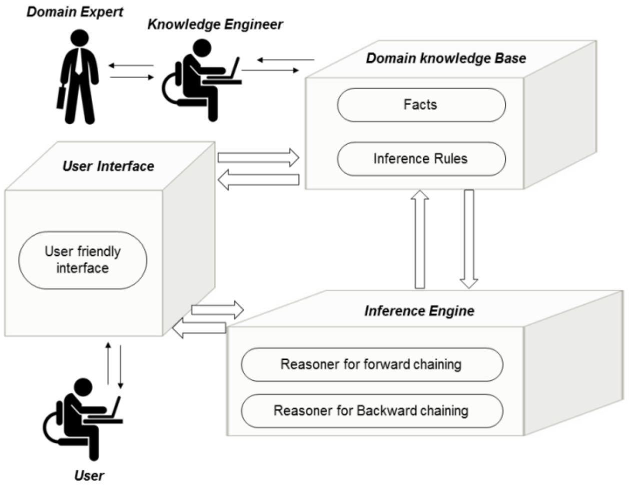


Fig. 1. Overall architecture of the proposed model, StuA.

The overall architecture of the proposed model is presented in Fig. 1. In the model, the knowledge engineer designed the knowledge-base in collaboration with the domain expert. The knowledge is stored in the form of facts and inference rules. The inference engine works with forward chaining as well as backward chaining. A user-friendly interface is provided to the user where he interrogates. To simplify the process, some drop downs are provided. The user asks for a query using the user interface. It is then treated by the inference engine. At this stage, first, the type of the query is analyzed. Depending upon the type of the query inference mechanism is carried out either through facts directly, or through forward chaining or the backward chaining using the inference rules. The result is then displayed to the user. Unlike CLIPS, only the relevant rules are fired and the filtered and processed output is displayed. The filtering and processing of output are being carried out by integrating CLIPS with java.

A. Domain Knowledge-Base

Domain specific knowledge is considered as bottleneck information in building knowledge-based systems. The model is trained with minimum possible knowledge. It often encountered missing data in running new test examples thus using inference mechanism it is able to answer them. Here, Information is handcrafting which is the simplest way to put knowledge into program. The focus is mostly on gathering knowledge from human experts, college website and through feedback from students.

One of the key points while constructing a virtual assistant is transparency i.e., making the system understandable despite the complexity of task. This is because:

- The system improves through consecutive development, which requires thorough understanding of earlier versions.
- The system improves through criticism from the people who are not familiar with its implementation details.
- The system uses its own learning methods for solving the problems.

After acquiring knowledge from human experts, it maps the knowledge so that it can be used in the program. Knowledge is basically in the form of sentences which is then broken into subject, object and predicate so that they could be programmed in the knowledge-base.

B. Inference Mechanism

A survey was conducted to observe the type of query any newcomer could ask and four major classes of questions is identified. The first class is “TRUE/FALSE” which is used to tell whether an asked fact is true or not. For example “Student can return book on Monday “, such statement is either true or false. The second class of question is “WHAT-IS or WHERE”. This class means that these questions are related to some atomic knowledge about the environment. For example, we have some information like “LRC is learning resource center. The minimum CGPA is 4”, these are some of the atomic information which cannot be further broken. So the query can be “what-is minimum CGPA?” and this can be directly extracted from the information available. Similarly, the third class is “WHAT-HAPPEN”. This class identifies the consequences of some situation or helps to know the outcome of some event. Suppose we have some fact like “if student fails supplementary examination then they get back in that year”, in this fact query could be, “what happens if student fails supplementary examination?” This also led to the identification of the fourth class of question that is “WHEN” which is meant to know the cause for some event. For example, we have a fact that “if attendance is less than 60%, students get debarred.” In such case, the user can ask the query that “when student get debarred?” So, for all these situations of an environment, the queries are divided into various classes. All the queries can be broadly classified into these four classes only. However, some queries can also be resolved in one level inferencing while others may require deep reasoning. Both are handled in the model.

The proposed model has a rule and fact based inference system. In this model, the TRUE/FALSE class is directly handled by CLIPS. The information is stored as fact in Knowledge Base (KB) and to answer that the fact is true or not, KB is searched. The information handled by “WHAT-IS or WHERE” class is atomic knowledge and can be inferred directly thus they are stored as facts in the knowledge-base of our model. The “WHAT-HAPPEN and WHEN” class has the information stored in the form of rules. In our model, the rules are typically structured as antecedent with their consequents. The inference engine examines the type of class to which the query belongs and then responds accordingly by executing the corresponding consequents or by searching for the correct antecedent. Forward chaining approach is used to infer queries of WHAT-HAPPEN class as in this class we want to know the consequent of a situation. Forward chaining is an inferencing method. It uses the available data and inference rules to extract more data until it is able to find the goal. This method is also called Data-driven as the data determine which rules should be selected and used. For example, the following information is present in the KB:

Initial fact:

Grade is F

Rules:

If (grade is F) then (student fails subject).

If (student fails subject) then (has to take supplementary exam)

If we want to know "what-happen-if grade is F". For this, we need to search for the consequent with the help of forward chaining. Similarly, backward chaining approach (explained in section III.C) is used to answer the query of WHEN class. Here, the goal is to decide that can we infer the fact "when student has to take supplementary exam" from the initial facts or not? With the help of backward reasoning, it tries to prove the goal (i.e. student has to take the supplementary exam). Hence, we have to show that antecedent (i.e. student fails subject) can be proved. Now, this becomes our new sub goal and so on. This continues until the initial fact as the antecedent is found.

To accommodate all these types of queries, an inference engine is proposed which is made up of query processor and different types of processing models as shown in Fig. 2. This inference engine does all the inferencing. It selects correct processing model for a given query. There are four models for query processing i.e. True False (TF) processor, Fact processor, Conclusion processor and Backward Chaining (BC) processor. The TF processor answers query for the TRUE/FALSE class of query. The Fact processor extracts the fact from the belief base to provide an answer. It handles the WHAT-IS and WHERE class of queries. The Conclusion processor infers a fact from the knowledge-base on the basis of the query provided to the inference engine. This processor uses forward chaining for the processing and thus handles the WHAT-HAPPEN class of queries. The BC processor uses the backward chaining approach to infer the facts for answering the WHEN class of queries.

The model is implemented with help of CLIPS and JAVA. CLIPS tool is integrated with JAVA with help of CLIPSJNI [18]. This integration helped in overcoming some limitation of CLIPS. First of all, it suppressed the triggering of all rules. Secondly, it also suppressed the assertion of irrelevant facts. As a result, the response time is also improved and only the relevant information is shown to the user. Moreover, CLIPS doesn't have a feature of backward chaining also. So, using JAVA, CLIPS is extended with a generic form of backward chaining which can provide a first level explanation of the queries.

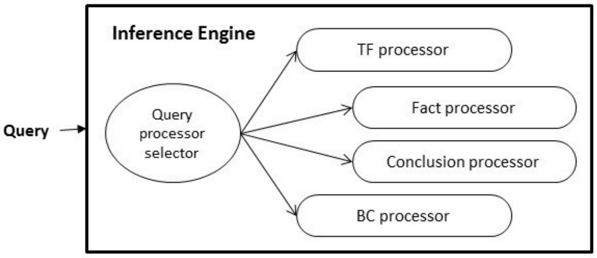


Fig. 2. Inference Engine of Proposed Model.

C. Extended CLIPS Tool

Backward chaining is an inference methodology in which a conclusion is available and moving backward to find the base facts supporting the conclusion is possible. CLIPS is a public domain software tool which is widely used for building intelligent systems [2-3]. It combines the programming paradigms of procedural, object oriented and logical languages, but it does not support backward chaining [24-25]. Backward chaining is important as it is a faster approach and it is more suitable for goal driven queries.

There are various queries that require backward chaining. For example, if someone asks "when they can get debarred" then there exists a rule "if attendance less than 60% then get debarred" in the knowledge-base. So here, in this case, given the consequent, antecedents should be determined. Such type of queries is handled with help of backward chaining. It is observed that mostly "WHEN" type of queries can be answered using backward chaining and they are being processed by BC processor module of the proposed model.

Algorithm I: backward_chaining

```
Input: Query
Output: Response

Declare environment variable clips1, clips2
Load facts in clips2
Load backward chaining rules in clips1
Add query fact in clips1
fact_list← run clips1
if fact_list length $\leq 1$
        ans← "nothing can be inferred"
end if
else
     connection_list← run clips1
     for each fact fiє fact_list do
          v← extract value from fi
          new_query← create query from v
          append v in bc_factlist
          fl2← run clips2
          if $fl2 \neq \phi$ then
               append v in delete_fact_list
          end if
     end for
end else

for each Di є delete_fact_list do
     if bc_factlist contains Di then
          remove from bc_factlist and clips1
     end if
end for

settingValue (first element of connection_list)

for each Fi є connection_list do
     if correct_flag of Fi is True then
          ans← fact connected to Fi
     end if
end for
```

Algorithm II: settingValue(connection_list element)

```
Input:connection_list
Output: correct_flag and check_flag for all elements in
connection_list are set accordingly

Declare correct_flag← true and check_flag← false
if check_flag of Fi is true then
    return
end if
If correct_flag of Fi is false then
    check_flag of Fi ← true
end if
for each Fi ϵ connected fact of Fi in connection_list do
    if bc_factlist not contains connected fact then
        H ← Search for connected fact of Fi in connection_list
        if H = $\phi$ then
            correct_flag← false and check_flag← true
            break loop
        end if
        else
            settingValue (H)
        end else
    end if
end for
```

In this algorithm, firstly, a set of basic facts is created that exists in our knowledge-base with help of bc_factlist variable. Also, a list named connection_list is used to maintain the relationship between different base and derived facts. In settingValue() function recursive call is used to successfully infer a set of facts for a given query. This is done by searching for a given fact in the base facts list. For a derived fact, a recursive call is made. Through this recursive call, the facts deriving the other facts are being searched in the fact-base list.

IV. Implementation

The proposed model, StuA, is implemented using JAVA and CLIPSJNI. A knowledge-base is created with various facts and rules related to a specific domain i.e. the college environment. It takes the query as an input. The query is provided as an input to inference engine which distinguishes the query class and sends it to the appropriate processor. The processor answers the query by forming a simple sentence using a first level of natural language generation. The user interface is shown in Fig 3(a)-3(d).

Fig. 3(a) shows the simulation of TRUE/FALSE class. It uses the TF processor to process the query. The TF processor searches if the fact asked exists in the knowledge-base or not accordingly answer the query. In Fig. 3(b), the query of WHAT-IS or WHERE class is being processed by Fact processor. It searches for facts matching the given information and extracts the rest of the information from there. It further uses that extracted information to answer. The WHAT-HAPPEN class of query is performed by conclusion processor as shown in Fig. 3(c). It processes the query by finding all the relevant facts that could be inferred. It uses forward chaining for this process. It finally sorts the inferred facts and answers the query. Fig. 3(d) shows the simulations related to WHEN class. This class of queries is processed using the BC processor in which the proposed backward chaining method is implemented. This method searches for the initial facts in the fact-base which could be reached from the goal statement and stores the first level information related to the particular goal. This first level information is processed to form a sentence if an initial fact is successfully found. Table II lists some questions posed to the virtual assistant, StuA.

TABLE II.
Sample Question Answers Processed By The Proposed Model, StuA

Question asked by user	Answers given by StuA
Where is LT1	Ground floor have LT1
What is min CGPA	Min. CGPA is 4.0
What happen student return book late in LRC	Student has to pay fine
When student get debarred	Attendance less than 60
What happen student fails supplementary exam	Student fails supplementary exam implies student get back , have to take summer semester, backlog
How can one access study material	Connect to college intranet and use the login details

V. Verification and Validation

The testing process of any automated tool includes verification and validation. Verification checks the completeness of the knowledge-base. Validation checks the correctness of the knowledge-base in terms of consistency. As suggested by Wentworth et al. [27] and Ghasem & Alizadeh [26], firstly the logical completeness and logical consistency of the knowledge-base is checked. Then, the knowledge model is validated followed by the validation of semantic consistency of knowledge items. Finally, the backward chaining algorithm integrated with CLIPS is validated.

A. Logical Completeness

Logical completeness means the expert system produces some conclusion for all inputs. This can be done by following the below mentioned steps [27]:

1. Constructing a logical formula that represents conditions under which the system is complete; this logical formula will be called the completeness formula in conjunctive normal form.
2. Eliminate ORs containing logical opposites or all possible values of a variable.
3. If the resulting logical expression is TRUE, the system is complete.

We checked the completeness of all the subsystems as specified above and found them COMPLETE. For illustration, Study_Material_Accees subsystem completeness check is shown below:

To show the completeness of Study_Material_Accees subsystemof Knowledge Base, firstly construct the logical formula:

Study_Material_Accees = YES

OR Study_Material_Accees = NO

Expressing in terms of input conditions:

(Connected_internal_Network= YES

andHave_Login_Access = YES)

OR

(Connected_internal_Network= No

OR Have_Login_Access = No)

Writing this in conjunctive normal form gives

(Connected_internal_Network= YES

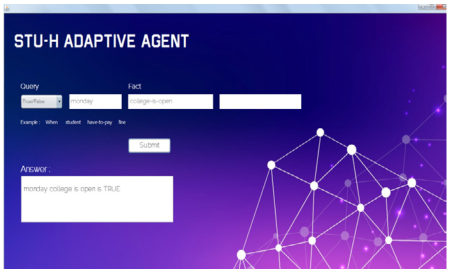
OR Connected_internal_Network= No

OR Have_Login_Access = No)

AND

(Have_Login_Access = YES

OR Connected_internal_Network= No



STU-H ADAPTIVE AGENT

Query: True/False | monday

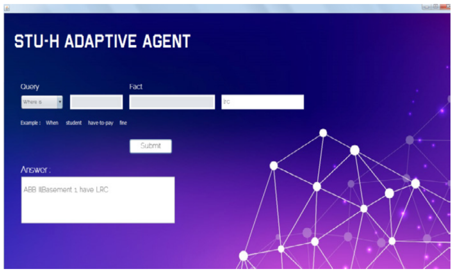
Fact: college-is-open

Example : When student have-to-pay fine

Answer :

monday college is open is TRUE

Fig. 3(a). Simulation of TF processor.



STU-H ADAPTIVE AGENT

Query: Where is

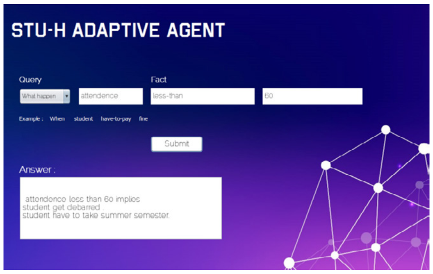
Fact: lrc

Example : When student have-to-pay fine

Answer :

ABB IIBasement 1 have LRC

Fig. 3(b). Simulation of Fact processor.



STU-H ADAPTIVE AGENT

Query: What happen | attendence

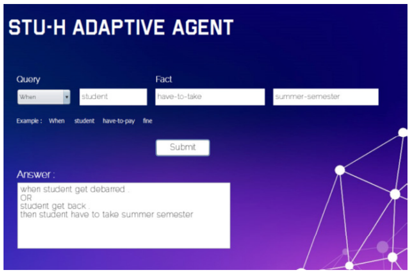
Fact: less-than | 60

Example : When student have-to-pay fine

Answer :

attondonco loss than 60 implios
student get debarred .
student have to take summer semester.

Fig. 3(c). Simulation of Conclusion processor.



STU-H ADAPTIVE AGENT

Query: When | student

Fact: have-to-take | summer-semester

Example : When student have-to-pay fine

Answer :

when student get debarred .
OR
student get back .
then student have to take summer semester

Fig. 3(d). Simulation of BC processor.

OR Have_Login_Access = No)

The first term is TRUE because yes and no are the only possible values for Connected_internal_Network. Likewise the second term is also TRUE. Therefore, the formula expressing completeness of Study_Material_Accees is TRUE, and hence the subsystem is complete.

B. Logical Consistency

Logical consistency means for all inputs, the knowledge base produces a consistent set of conclusions, i.e., that for each set of possible inputs, all the conclusions can be true at the same time. To establish consistency, the user must do the following [27]:

1. Construct a logical formula that represents conditions under which consistency fails; this logical formula will be called the consistency formula. Write this formula in disjunctive normal form.
2. Eliminate ANDs containing logical opposites or other contradictory sets of conjuncts.
3. If the left hand side of the resulting logical expression is FALSE, the system is consistent.

We checked the consistency of all the subsystems as specified above and found them CONSISTENT. For illustration, Study_Material_Accees subsystem consistency check is shown below:

To show the consistency of Study_Material_Accees subsystem of Knowledge Base, firstly construct the logical formula. The only set of inconsistent conclusions is

Study_Material_Accees = YES

AND Study_Material_Accees = NO

Expressing in terms of input conditions:

(Connected_internal_Network= YES

andHave_Login_Access = YES)

AND

(Connected_internal_Network= No

OR Have_Login_Access = No)

Writing this in conjunctive normal form gives

(Connected_internal_Network= YES

AND Have_Login_Access = YES

AND Connected_internal_Network= No)

OR

(Connected_internal_Network= YES

AND Have_Login_Access = YES

AND Have_Login_Access = No)

*The first term is FALSE because yes and no are contradictory values for Connected_internal_Network. Likewise the second term is also FALSE. Therefore, the formula expressing consistency of Study_Material_Accees is FALSE, and hen*ce the subsystem is consistent.

C. Knowledge Models Completeness Check

Logical completeness and consistency are necessary but not sufficient for a knowledge model to be complete. It should be *semantically complete* as well, i.e., it must base its decisions on all information considered to be relevant by the expert [27].

One of the ways to check the completeness of a knowledge model is to create a knowledge model with a single expert and review the knowledge model with other experts who are not connected with the development of the model. Following the same, the knowledge model is created by knowledge engineers and for completeness check, we selected 70 final year students (46 males and 24 females) of the college as they seem to be the best suited experts in this domain. They were asked to use the tool for one whole day and check for the correctness of the answers given by the virtual assistant. All the 70 students, in total, posed 836 questions to the assistant and found 829 answers correct. All of them were satisfied with the working of the tool and 99.16% accuracy is reported.

D. Validating the Semantic Consistency of Underlying Knowledge Items

Even if the expert knowledge has been properly encoded into an expert system knowledge-base, the KB will probably produce errors if the underlying expert knowledge is wrong. Therefore, it is important to validate the expert knowledge behind the knowledge-base. This can be done by checking the confidence level of experts.

The basic method for validating a knowledge item is [27]:

- Ask a panel of experts whether it is true or false.
- Tally the TRUE/FALSE answers.
- Analyze the results statistically.

This test is conducted on the same population of experts. All the 70 students were asked to answer 35 question (excerpts shown in Table III) as yes or no. All the experts agreed upon all the 35 questions unanimously. The confidence level is computed using the formula

Confidence Level = 1 - (1 / 2**N)

Where N is the number of experts. In this experimentation, 100% confidence is reported.

TABLE III. Excerpts from True/False Questionnaires

Questions	T/F
If student has not given T1 or T2, he can give makeup exam	True
If student is debarred, he can give supplement exam	False
If student is debarred, he can appear for summer semester	True
If student fails, he can give supplement exam	True
If CGPA is less than 4, student get year back	True
If students returns book late in LRC, he need not to pay fine	False
100% weight-age is given in makeup exam	False

E. Backward Chaining Algorithm Validation

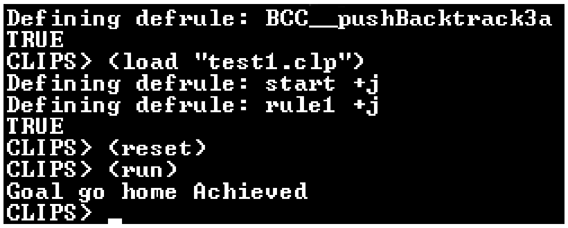
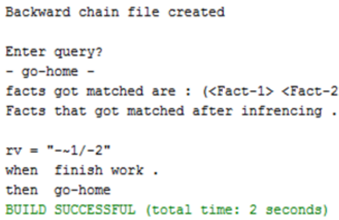
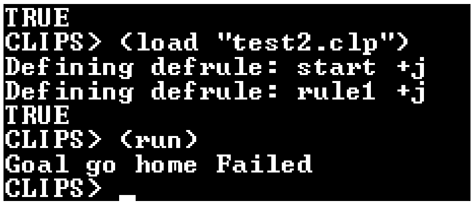
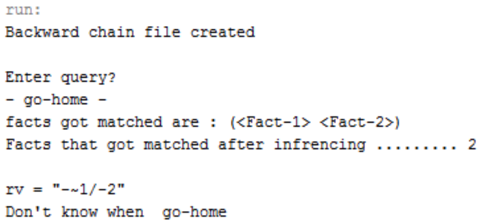
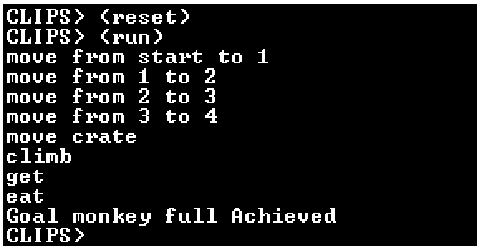
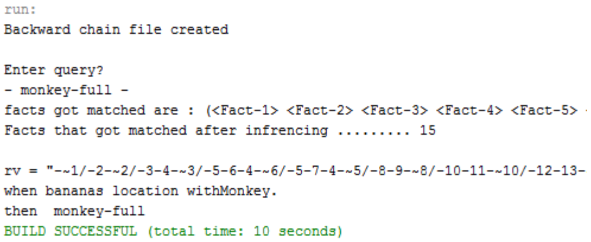
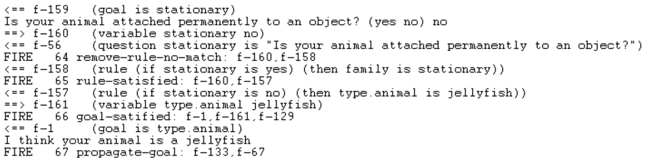
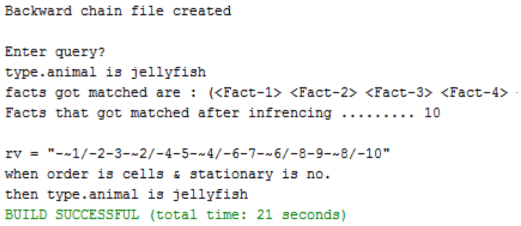
As a next step of validation, the model is simulated for the pre-simulated domains [22] and [23]. In this step, mainly backward chaining module was validated. A few existing models have implemented the domain specific backward chaining. To test our generic model, we used those simulations. Table IV shows the testing results of the proposed model for backward chaining design in CLIPS with the help of JAVA. In table IV, the output of some of the pre-simulated examples of backward chaining is compared with our proposed model. Through this testing, it is concluded that the proposed model works perfectly in various scenarios. 100% correctness for all the examples is achieved and with an advantage of providing single level explanations.

VI. Conclusion

People sometimes hesitate to interact with a stranger and ask their queries in a new environment. A virtual assistant provides a solution for this. In this paper, we proposed an interactive and intelligent student assistant, StuA, situated in a specific domain (i.e. the college environment) where it is capable of answering all types of queries of a new-comer to make him/her familiar with the new environment. It facilitates real-time, low-cost expert-level assistance with 24X7 availability. The model is designed using CLIPS as it allows inferencing. Further, it provides a one-level explanation of queries, which is its advantage over other existing models. We have worked on the limitation of CLIPS by proposing and implementing a generic model of backward chaining. The model is validated on various pre-simulated examples which gave 100% correctness. Further, successful checks are performed for logical completeness and consistency of the knowledge-base. Further, semantic completeness is also checked with the help of 70 domain experts and found the accuracy of 99.16%

The proposed system is restricted to the domain of college and

Table IV. Comparison of Existing Specific Model with the Proposed Generic Model

Pre-simulated scenarios Results	Results generated by the proposed model
 Defining defrule: BCC__pushBacktrack3a TRUE CLIPS> (load "test1.clp") Defining defrule: start +j Defining defrule: rule1 +j TRUE CLIPS> (reset) CLIPS> (run) Goal go home Achieved CLIPS> _	 Backward chain file created Enter query? - go-home - facts got matched are : (<Fact-1> <Fact-2>) Facts that got matched after infrencing .... 2 rv = "-~1/-2" when finish work . then go-home BUILD SUCCESSFUL (total time: 2 seconds)
 TRUE CLIPS> (load "test2.clp") Defining defrule: start +j Defining defrule: rule1 +j TRUE CLIPS> (run) Goal go home Failed CLIPS> _	 run: Backward chain file created Enter query? - go-home - facts got matched are : (<Fact-1> <Fact-2>) Facts that got matched after infrencing .... 2 rv = "-~1/-2" Don't know when go-home
 CLIPS> (reset) CLIPS> (run) move from start to 1 move from 1 to 2 move from 2 to 3 move from 3 to 4 move crate climb get eat Goal monkey full Achieved CLIPS>	 run: Backward chain file created Enter query? - monkey-full - facts got matched are : (<Fact-1> <Fact-2> <Fact-3> <Fact-4> <Fact-5> Facts that got matched after infrencing .... 15 rv = "-~1/-2-~2/-3-4-~3/-5-6-4-~6/-5-7-4-~5/-8-9-~8/-10-11-~10/-12-13- when bananas location withMonkey. then monkey-full BUILD SUCCESSFUL (total time: 10 seconds)
 <== f-159 (goal is stationary) Is your animal attached permanently to an object? (yes no) no ==> f-160 (variable stationary no) <== f-56 (question stationary is "Is your animal attached permanently to an object?") FIRE 64 remove-rule-no-match: f-160,f-158 <== f-158 (rule (if stationary is yes) (then family is stationary)) FIRE 65 rule-satisfied: f-160,f-157 <== f-157 (rule (if stationary is no) (then type.animal is jellyfish)) ==> f-161 (variable type.animal jellyfish) FIRE 66 goal-satified: f-1,f-161,f-129 <== f-1 (goal is type.animal) I think your animal is a jellyfish FIRE 67 propagate-goal: f-133,f-67	 Backward chain file created Enter query? type.animal is jellyfish facts got matched are : (<Fact-1> <Fact-2> <Fact-3> <Fact-4> Facts that got matched after infrencing .... 10 rv = "-~1/-2-3-~2/-4-5-~4/-6-7-~6/-8-9-~8/-10" when order is cells & stationary is no. then type.animal is jellyfish BUILD SUCCESSFUL (total time: 21 seconds)

academics which can be customized to various other domains such as at reception of hotels, hospitals, and offices or in schools, malls or in an organization. The proposed model of backward chaining can further be extended to incorporate loops and other higher level programming components. This paper opens lots of future possibilities. A simpler query-writing mechanism can be designed, which might be achieved through Natural Learning Processing (NLP). Nevertheless, it can be made more adaptive by adding learning. With learning and continuous user interaction, it can become more efficient in answering queries.

References

[1] E. A. Feigenbaum and P. McCorduck, The Fifth Generation, Addison-Wesley, 1984.

[2] J. C. Giarratano, G. Riley, Expert Systems: Principles and Programming, 4th Ed, Course Technology, 2004.

[3] J.C. Giarranto, CLIPS User's Guide, Version 6.22, 1998.

[4] T. Sharma, N. Tiwari and K. Shah, "Student counselling system: A Rule-based expert system based on Certainty factor and backward chaining approach," International Journal of Application or Innovation in Engineering and Management, vol. 2, no. 1, pp. 371-375, 2013.

[5] Y. Bouaiachi, M. Khaldi and A. Azmani, "A Prototype Expert System for Academic Orientation and Student Major Selection," International Journal of Scientific & Engineering Research, vol. 5, no. 11, pp. 25-28, 2014.

[6] Al-Ghamdi, S. Al-Ghuribi, A. Fadel, F. Al-Aswadi and T. AL-Ruhaili, "An Expert System for Advising Postgraduate Students," International Journal of Computer Science and Information Technologies, vol. 3, no. 3, pp. 4529-4532, 2012.

[7] S. S. A. Samy, M. Baraka and A. Baraka, "A Proposed Expert System For Guiding Freshman Students In Selecting A Major In Al-Azhar University, Gaza", Journal of Theoretical and Applied Information Technology, vol. 4, no. 9, pp. 889-893, 2008.

[8] M. Lazzari and P. Salvaneschi, "MISTRAL: an Expert System for the management of warnings from automatic monitoring systems of dams," Proceedings of the 2nd Specialist Seminar on integrated CAD of earthquake resistant buildings and civil structures, 1993.

[9] D. Selva, "Rule-based system architecting of Earth observation satellite systems", PhD dissertation, Massachusetts Institute of Technology, Cambridge, MA, US, 2012.

[10] D. Selva, B. Cameron, and E. F. Crawley, "A rule-based method for scalable and traceable evaluation of system architectures", Research in Engineering Design, vol. 25, no. 4, pp. 325–349, 2014.

[11] S. S. A. Naser, A. Z. A. Ola, "An expert System for Diagnosing eye disease using CLIPS," Journal of Theoretical And Applied Information Technology, vol. 4, no. 10, pp. 923-930, 2008.

[12] Hofmeister, M. Alan, Althouse, R.B. Likins, M. Morgan, D.P. Ferrara, J.M. Jenson, W.R. Rollins and Evelyn, "SMH.PAL: an Expert System for identifying treatment procedures for students with severe disabilities," Exceptional Children, vol. 61, no. 2, pp. 174-181, 1994.

[13] S. S. A. Naser, K. Kashkash and M. Fayyad, "Developing an Expert System for Plant Disease Diagnosis," Journal of Theoretical and Applied Information Technology, vol. 1, no. 2, pp. 78-85, 2008.

[14] A.T. Al-Taani, "An Expert System for Car Failure Diagnosis," Proceedings of World Academy of Science, Engineering and Technology, vol. 7, pp. 457-460, 2005.

[15] R. Borgohain, S. Sanyal, "Rule-based Expert System for Diagnosis of Neuromuscular Disorders," International Journal of Advanced Networking and Applications, vol. 4, no. 1, pp. 1509-1513, 2012.

[16] K. Robindro, S. K. Sarma, "JESS based expert system architecture for diagnosis of rice plant disease: design and prototype development," 4th International Conference on Intelligent Systems, Modelling and Simulation, pp. 674-676, 2013.

[17] B. M. Salem, M. Roushdy and R. A. Hod, "A case based expert system for supporting diagnosis of heart diseases," AIML Journal, vol. 5, no. 1, pp. 33-39, 2005.

[18] S. A. Samy, R. Al-Dahdooh, A. Mushtaha and M. El-Naffar, "Knowledge Management in ESMDA: Expert System for Medical Diagnostic Assistance," AIML Journal, 2010.

[19] S. A. Samy and S. Almursheidi, "A Knowledge Based System for Neck Pain Diagnosis," World Wide Journal of Multidisciplinary Research and Development(WWJMRD), vol. 2, no. 4, pp. 12-18, 2016.

[20] Y. A. Nada, "A Novel Expert System for Building House Cost Estimation: Design, Implementation, and Evaluation," International Journal of Advanced Computer Science and Applications, vol. 4, no. 8, pp. 50-55, 2013.

[21] N. L. Griffin, F.D. Lewis, "A Rule-based Inference Engine Which is Optimal and VLSI Implementable," IEEE International Workshop on Tools for Artificial Intelligence, VA, pp. 246-251, 1989.

[22] J. C. Giarratano, CLIPS user's guide, Artificial Intelligence Section, Johnson Space Center, 1987.

[23] Huyck, http://www.cwa.mdx.ac.uk/bis2040/backchain/backchain.html, dated May, 2017.

[24] S.H. Liao, "Expert system methodologies and applications—a decade review from 1995 to 2004," Expert Systems with Applications an International Journal, vol. 28, no. 1, pp. 93-103, 2005.

[25] Kishan, V. Chadha and C. Maini, "A Review of Development and Applications of Expert," International Journal of Advanced Research in Computer Science and Software Engineering, vol. 2, no. 10, pp. 319-325, 2012.

[26] Armin Ghasem Azar, Zohreh Mohammad Alizadeh, "A Basic Proof Method for the Verification, Validation and Evaluation of Expert Systems," Journal of Information Systems and Telecommunication, vol. 1, no. 1, pp. 27-31, 2013.

[27] James A Wentworth, Rodger Knaus, Hamid Aougab, "Verification, Validation, and Evaluation of Expert Systems", An FHWA Handbook, Volume 1, 1997.

[28] Bhattacharya, S., and S. Nath. "Intelligent e-Learning Systems: An Educational Paradigm Shift," International Journal of Interactive Multimedia and Artificial Intelligence, vol. 4, no. 2, pp. 83-88, 2016.

[29] Secades, V. A., and O. Arranz. "Big Data & eLearning: A Binomial to the Future of the Knowledge Society," International Journal of Interactive Multimedia and Artificial Intelligence, vol. 3, no. 6, pp. 29-33, 2016.

[30] Jain, S. and Asawa, K., "Formalization of Event Perception and Event Appraisal Process," International Journal of Interactive Multimedia and Artificial Intelligence, vol. 3, no. 3, pp. 69-73, 2015.

[31] Choudhar, R.S., Kukreja, R., Jain, N. and Jain, S., "Personality and Education Mining based Job Advisory System," International Journal of Interactive Multimedia and Artificial Intelligence, vol. 2, no. 7, pp. 30-34, 2014.

Pooja Lodhi

Pooja Lodhi is pursing M.Tech (Computer Science) from Jaypee Institute of Information Technology (JIIT), Noida, India. Her area of interest includes Artificial Intelligence, Agent Modelling, Expert Systems and Machine Learning.

Omji Mishra

Omji Mishra is pursing M.Tech (Computer Science) from Jaypee Institute of Information Technology (JIIT), Noida, India. His area of interest includes Expert Systems, Soft Computing and Machine Learning.

Vasvi Bajaj

Vasvi Bajaj is a undergraduate student at Jaypee Institute of Information Technology (JIIT), Noida, India. Her area of interest includes Machine Learning and Artificial Intelligence.

Shikha Jain

Shikha Jain is presently working with Jaypee Institute of Information Technology (JIIT), Noida, India as Assistant Professor. She has more than fourteen years of research and academic experience. She has received her PhD in computer science from Jaypee Institute of Information Technology, Noida, India, in the area of "Emotion modelling for an autonomous agent and its programming". She has published a number of research papers in the renowned journals and conferences. Her research area includes Affective Computing, Emotion Modelling, Cognitive Affective Architectures, Agent Modelling, Machine Learning, Soft Computing, Algorithm Design and Automata theory. She is also member of technical program committee of several conferences.