Asian Journal of Engineering and Technology Innovation

ISSN : 2347-7385

REVIEW ARTICLE

Received on: 1-11-2014
Accepted on: 15-11-2014
Published on: 19-11-2014

Hardik patel, Swati verma , Inshu rani, Abhilasha bhalke
Computer Engineering Dr. D.Y.P.C.O.E, Ambi,Talegaon-Dabhade Maharashtra, India
hrpatel163@gmail.com
swativrm.rocks15@gmail.com
inshurani99@gmail.com
bhalkeabhi789@gmail.com

QR Code for Mobile users

Conflict of Interest: None Declared

Automation of HR Interview System Using 'Jess' Inference System

Hardik patel, Swati verma , Inshu rani, Abhilasha bhalke
Computer Engineering Dr. D.Y.P.C.O.E, Ambi,Talegaon-Dabhade

ABSTRACT
Inference Engines are software that derive conclusions out of existing data and also fire actions if the activity trend gets a match within existing knowlegde base. Inference engine hence are capable and taking decisions base on there heuristics and learning. The fact can be harness into developing intelligent software system which their accuracy with time and become more dexterous in handling complex situation. This software deals with the design and development of an application automating HR Interview process, base on one of the popular Inference Engine i.e. JESS, it is demonstrating the power of Inference Engines to satisfy the need of complex logical requirements.

Keywords: jess, inferencesystem, expert system.

Cite this article as:
HARDIK PATEL, SWATI VERMA , INSHU RANI, ABHILASHA BHALKEAUTOMATION INTERVIEW SYSTEM USING 'JESS' INFERENCE SYSTEM, Asian Journal of Engineerin Technology Innovation 02 (05); 2014; 26-29.

INTRODUCTION

Inference engine is a software tool which is use to learn new thing which is already existing knowledge by using rules, called as inference rules. The Inference Engine is very much similar to F.S.M i.e. Finite State Machines with a cycle consisting of three action states:Match Rules, Select Rules, and Execute Rules. The first state, 'Match Rules' involves Inference Engine finding all of the rules that are satisfied by the current contents of the data store. The rules in typical condition-action form, indicates the testing of the conditions against the working memory. The rule matching provides necessary input for execution, collectively referred as the conflict set. The appearance of the same rule several times indicates the conflict set, if it matches different subset of data item. The pair of a rule and a subset of matching data item are called instantition of the rule. The conflict set is now passed on to the second state that is 'Select Rules' by the inference engine. The select state involves selection of strategy by the inference engine in order to determine which rule will actually get executed. The selection strategy can be hard-coded into the engine or may be specified as part of the model. In the larger context of Artificial Intelligence, these selectionstrategies are often referred as heuristics. The selected instantiations along with the selected rules are passed over to the third state 'Execute Rule'. The inference engine executes or fire the selected rules, with the instantiations data item as parameter.

What is JAVA EXPERT SYSTEM SHELL(JESS)?

JESS is a rule engine and scripting environment written entirely in Java language by Ernest Friedman-Hill at Sandia National Laboratories in Canada. JESS is originally inspired by the CLIPS expert system shell, but havegrow into a complete, distinct Java influence environment of it own. JESS can be used to build Java applets and applications with the capacity to "reason" using knowledge supplied in the form of declarative rules. The syntax for JESS is very much similar to CLIPS and Lisp. JESS is used for creating intelligent software called expert system. An expert system have a set of rule that are applied repeatedly to a collection of facts about the world. It is a specifically intended to model human expertise or knowledge. For matching of rules to facts, jess make use of rete algorithm.

There are three way to represent knowledge in JESS:

(i) Rules, which are primarily intended for heuristic knowledge based on experience.

(ii) Functions, which are primarily intended for procedural knowledge.

(iii) Object-oriented programming

i.e OOP, also primarily intended for procedural knowledge. It allow pattern to match of Rules on objects and facts.

WORKING:

The initial focus of the paper was to automate a process which would require core intelligence on the part of the software application. The aim of the work was to build an application to be practically usable and one that is difficult to realize using normal programming languages. It was thus decided that a HR interview application can be chosen as a standard application subject to exemplify the capabilities of JESS. Over the years of industrial workforce research we are quite aware that talent is simply not enough without the strong willingness and devotion towards work. HRinterview is done to decide how much a person is mentally prepared, devoted and how good are his work ethics. HR interview is an example of an activity which relies heavily on the psyche of the interviewer and the interviewee. Not every question asked and every answer provided can be hard coded to represent them as perfectly right or perfectly wrong. And each question asked has something to do with the previous or the next question. The interviewee can't expect to pass through a particular question by just giving a correct answer to it without actually answering correct each related question perfectly, for example : Question asked : Why do you want to join this company ? Answer provided: Because I am really interested in the field the company works in. Now the answer may seem quite ok and a normal application would pass the candidate in this question but an expert system waits for other related questions to decide about the result of this question, for example : NextQuestion asked : What does this company works in ? Answer provided: I don't know. Now this answer clearly says that the interviewee didn't actually give the real reason behind joining the company. Hence both answers are marked as insufficient. Next complication dealt was, human evaluations are not discreet; they are attached with terms like almost, not quite and too some extent. Hence application's evaluation process was made to award variable marks to each answer not just judge them as right or wrong.

The Main concepts which we are using in this project.

1. The Concept of Minimum Marks.
2. The Concept of Total Minimum Marks.
3. The Concept of Impression.
4. The Concept of Mutation.

IMPLEMENTATION DETAILS

The previous section dealt with concept formulation which helps in realizing the internal details and working of the proposed application. The next steps involve rule formulation using JESS. The concepts were realized into JESS codes as discussed below:

(i) The Concept of Minimum Marks:

Choosing an unacceptable option as answer can lead to sudden termination. Rule below demonstrates how we can set a condition like, if an applicant answers a particular option of a particular question, he will be rejected immediately.

```
(defrule R_23 (answer (ident 4) (text ?t&:(eq ?t c))) =>
(printout t "Thank you for attending a interview, you are dis-sleceted !" crlf))
```

Rule above says, if a candidate answers 'c' option for question number '4', it should lead to rejection of that candidate.

(ii) The Concept of Minimum Total Marks:

In order to qualify the interview, a minimum score must be obtained or else it may lead to failure. Let us say a HR questionnaire has 18 questions, answers to each questions which are stored in variables 'a' through 'r' respectively, then below is how we can apply Concept of Minimum marks to it.

```
(defrule R_80 (test (< (+ ?a ?b ?c ?d ?e ?f ?g ?h ?i ?j ?k ?l
?m ?n ?o ?p ?q ?r) min_total_score)) =>
(printout t "Thank you for attending a interview, we are
sorry to inform that you are dis-selected !" crlf))
```

Rule above checks if sum of scores which are stored in variables 'a' through 'r' is greater than 'min_total_score' or not, otherwise it will lead to rejection.

(iii) The Concept of Partial Correction:

Every option has some associated mark. To have this implemented in our application we assigned marks to each answer option of a question. Rules below demonstrates how we can assign marks to an option of a question

```
(defrule R_56 (answer (ident 1) (text ?t&:(eq ?t a))) =>
(assert (marks (ident 1)(number 10))))
(defrule R_57 (answer (ident 1) (text ?t&:(eq ?t b))) =>
(assert (marks (ident 1)(number 5))))
```

First rule of above two rules says that if a candidate answers option 'a' for question number 1, he will be awarded with 10 marks. Second rule states that if a candidate answers option 'b' for question number 1, he will be awarded 5 marks. Similar rules can be written for each option answer
of every question.

(iv) The Concept of Impression:

Associated questions were awarded with advantage marks. Rules below demonstrates how we implemented The Concept of Impression in our application

```
(defrule R_72 (answer (ident 6) (text ?t&:(eq ?t b))) =>
(assert (marks (ident 3)(adv_number 4))))
```

Rule above says that, if a candidate answers option 'b' for question number 6, he will be awarded 4 extra bonus marks for question number 3.This is called a case of positive
impression.

```
(defrule R_73 (answer (ident 6) (text ?t&:(eq ?t a))) =>
(assert (marks (ident 10)(adv_number - 5))))
```

Rule above says that, if a candidate answers option 'a' for question number 6, he will be deducted 5 marks for question number 10. This is called a case of negative impression.

(v) The Concept of Mutation:

This can be triggered after a certain number of interviews have been completed. Rule below is one such rule, which checks the cumulative scored marks after 5 interviews get completed, and raise the threshold of minimum total score marks for a candidate to 140 from its old value. This change could either be an increased threshold limit or a decreased threshold limit, depending upon the interview scores till the point of time this rule is set to trigger.

```
(defrule R_90 (test (>cumulative_no 700))
=> (assert (marks (mini_total_score 140))))
```

CONCLUSION:

Inference Engines are very vast and new area of computer science with capabilities yet far from being even partially explored. We in our effort on this project have tried to unravel the capabilities of JESS inference engines.

Our model application was built for the very purpose of demonstrating the strength of expert systems based on JESS inference engine. Our application has been a bit limited in our approach due to scalability and complexity problems. We used very less number of interview questions and they were not changed during the interview. Still our applicationprovided an insight into how human logic can be imitated and how this benefit can be harnessed to build expert systems. We also demonstrated various inferencing techniques and presented the various programming logic involved while building intelligent software. We expect our effort would prove useful to anyone who wants to explore the capability of JESS inference engines to build even more complex software or anyone who simply wants to unravel the practical techniques of building an expert system. We hope to extend our project to make it commercially useful and applicable in diverse, practical environment.

FUTURE PROSPECT

With the ever changing requirement and expectation from software we are moving more close towards a reality where we would like to have intelligent software's that can improve themselves and adapt to the ever changing environment. JESS is really a strong platform for developing a sort of self-caring software that best imitates human intellect and inference drawing capability. In areas where human impression and feelings are involved can be automated by carefully designed expert systems. They can emulate a sort of human presence and make an user comfortable and not be too formal to fill in for the absence of intellect on the part of the software. We expecting to further enhance the application and even extend it to include more diverse areas where the application of JESS is justified.

REFERENCE

1. Arun N Nambiar, Anish.K. Dutta ," Expert System for Student advising using JESS" International Conference on Education and Information Technology, Pp. V1 312- V1 315, 2010
2. Swapna Singh, RaginiKarwayun "A Comparative Study Of Inference Engines" Seventh International Conference on Information Technology, Pp. 53-57, 2010
3. Ali Asgary, Albert Kong "Fuzzy-JESS expert system for indexing business resiliency", IEEE Toronto International Conference Science and Technology for Humanity , Pp.153-158, Sept. 2009
4. James P. McGlothlin, Latifur R. Khan: "RDFKB: efficient support for RDF inference queries and knowledge management, International Database Engineering and Applications Symposium (IDEAS 2009), Pp.259-266, Sept. 2009
5. Ernest Friedman-Hill Sandia "Jess The Rule Engine for the Java Platform", DRAFT Version 7.1p2, Nov. 2008
6. Ken KaLun Ho, Meiliu Lu "Web-based Expert System for Class Schedule Planning Using JESS", IEEE International Conference on Information Reuse and Integration, Pp.166-171, 2005
7. GizemOlgo "Inference Engines-Semantic Web Cross UpProject", July 2004
8. Maarten Menken " Jess Tutorial " Dec. 2004
9. Franz Baader, Werner Nutt " Basic Description Logics" Pp.43- 95. Description Logic Handbook 2003
10. G. Naudts, "An inference Enginunturk and Lei Zhang