Annales UMCS Informatica AI 7 (2007) 163-170

Annales UMCS
Informatica
Lublin-Polonia
Sectio AI

http://www.annales.umcs.lublin.pl/

Rule languages used in Tutorial chatbots programming

Barbara Gocłowska*, Michał Kufel, Monika Puchacz, Radosław Świech

Institute of Computer Science, Maria-Curie Sklodowska University,
pl. M. Curie-Skłodowskiej 1, 20-031 Lublin, Poland

Abstract

The function performed by the bots in Intelligent Tutorial Systems has been presented in the paper. Therefore we were able to justify the character of functions of the chatbot being designed by our team. Its purpose is to improve the functionality of an intelligent tutorial system. By means of tools used for the construction of the chatbot we were able to apply the strategies and techniques already used by commonly known authors as well as individual methods adapted to the ideas created in our group.

1. Introduction

The bots are used as the programs for data searching and acquisition. Most frequently 6 types of bots are used i.e. chatterbots, searchbots, shoppingbots, databots, updatebots, infobots according to the classification applied in IT literature [1]. The chatterbots, sometimes called chatbots simulating the dialog with the user, most frequently are used to obtain the knowledge from that user. That knowledge should be interpreted as meta-knowledge disclosing the gaps in user's knowledge and used as the source of information for a chatbot about the resources to be transferred to the user in the course of dialog and about the subjects of dialog to be selected. That role is reversed to the first role mentioned above, but is not conflicting with knowledge acquisition. In most intelligent systems such a dialog is used as a system supplementing tool and activated when any drastic gaps in user's knowledge are detected by the agent in the case of significant importance of the terms or associated problems (in our system such importance is called critical weight). The roles performed by chatbots in the Intelligent Tutorial Systems are diversified according to the system operation features and methods of training used in that system. However, their basic role is always the continuous acquisition of knowledge from the user in the course of

*Corresponding author: *e-mail address*: gocbar@gmail.com

the dialog with that user. In some systems [2-5], the chatbot performs the role of principal tutor verifying the knowledge of the user, giving the prompts, analysing the answers, searching and correcting the errors, presenting correct solutions and leading the dialog in order to obtain the answers to the specific questions. The chatbot in such systems performs the role of teacher, helper and the examiner simultaneously. Also the visual features of the chatbot – teacher have been provided by the authors of Auto Tutor system. The tutoring in that system is carried out in the form of conversation with a speaking head changing its mimics, gestures and tone of voice. The intention of the system authors was to make the learning process with the Auto Tutor system alike the lessons with the real teacher responding to the essential errors in the form of the look of His/ Her face and in the form of the special gesture in the case of progress.

In most Intelligent Tutorial Systems, the role of chatbots is limited to the organization of help system for the user. In such cases, the decision on chatbot calling is made by the user when any help is needed [6] or the chatbot is called automatically if certain conditions defined by the system are met [7]. After calling, the chatbot again plays the role of teacher – correcting the errors, asking the questions, demonstrating the correct solutions. However, when the problem is resolved and the user is provided with any expected or needed help, the chatbot withdraws itself to the background and the previous learning process according to the procedures preceding its calling is continued.

The chatbots are also applied in the Learning Companion System (LCS) [8], performing the Learning Companion function in the education process. The chatbot in the system may be applied as the model to be used by the user as the partner for cooperation and competition, as the user's learner or as the source of advice and instructions. In all those „creations", the task of the chatbot is to improve the efficiency of learning of its interlocutor by adaptation of learning method to His/ Her individual needs and preferences. Even two chatbots are used in the Duffy system [9] – one chatbot performing the role of the teacher and another used as the troublemaker acting as the Learning Companion giving correct answers as the suggestions for the learner or as the misleading Agent leading the user into error by means of wrong answers. The learning strategy using the troublemaker strategy is the solution being applied in the system in order to mobilize the student or to test His/ Her emotions as in the case of Duffy system.

The chatbots are more and more widely applied in the intelligent systems and proved in each phase of education process. The purpose of the learning strategies using the chatbots is to make their system being their environment and the learning procedures using that system alike the lessons with the private teacher. Not only the proper course of the dialog and the correct form of the phrases used by the chatbot are provided by the authors of Intelligent Tutorial Systems, but also its appearance, the look of face, tone of voice as well as its ability to induce the emotions of the user are considered more frequently. Increasingly often the chatbots are not used as a component of the Intelligent Tutorial Systems, but as the basis of their operation [3,4].

The chatboot ALICE [8] being commonly known among the Internet users is one of the most promising chatbots which has been awarded the Loebner Prize three times [10] (Turing Test [2]) so far. Despite its top position in that ranking, its structure is relatively simple. It consists of an interpreter and a knowledge base in the AIML language. The essential task of the interpreters is to search the knowledge base in order to adapt the answers to the present pattern in the form of user's statement. The AIML language (Artificial Intelligence Markup Language), created in order to satisfy the needs of ALICE, consists of four essential tags.

`<aiml>`: tag opening and closing the base
`<category>`: determining the part of knowledge being the focus area for ALICE
`<pattern>`: pattern
`<template>`: reply to pattern

Except for the tags presented above AIML contains about 20 other tags enabling the creation of more advanced knowledge base.

The answers to most questions or indicative sentences of the user are readily accessible in the knowledge base of ALICE program using an extremely comprehensive base. The patterns are provided in the form of ready complete sentences but also are associated with general statements indicated with an asterisk '*' or symbol.

2. The functions of educational chatbot in intelligent LISE system

According to our intentions, the aim of an educational chatbot is to teach the student who has been not credited with certain fragments of knowledge being limited but extremely important in view of the course coherence, associated with the determined part of the material exceeding the frame of the current course, for instance the knowledge which should have been gained by the student from the previous course. One could think that any creation of the chatbot for that purpose is unreasonable and that it would be sufficient to prepare the questionnaire asking the student which parts of material are unclear. However, the advantage of quasi – conversation consists in thorough understanding of the student's problem, „entering into" His/Her situation and prompting help. Furthermore any questionnaire is unable to present any context of the question originating in the student's brain, but several readily accessible answers depending on questions („Why", „What is it", „When" etc.) may be obtained from the chatbot in the course of conversation with the student. It can be realized in many ways; for example a sentence is subdivided into all possible fragments in the first phase by the chatbot searching for the answer patterns for each fragment thereafter. The most optimal answer is selected using certain principles (priorities, groups of rules, key words). In the course of analysis of the successive sentences given by the user, the conclusions drawn from the preceding dialogs and recorded in for global variables are considered by the chatbot.

JBossRules [12] rule engine has been used in the education chatbot implemented by our team as the result of the choice between AIML, JESS and readily available engines shells. The knowledge base consists of Jbossa rules. More detailed description of the structure of JBossRules rule engine has been included in further part of the present paper.

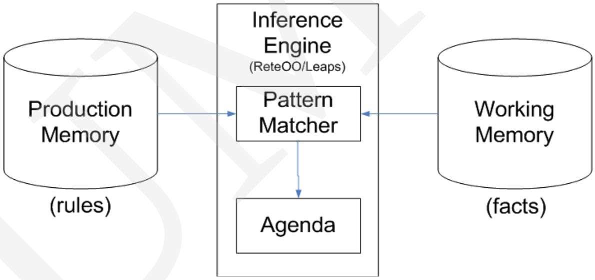

Fig.. 1. The structure of JBossRules rule engine

All rules are stored in the Production memory (in the form of files with extension drl). RETE or LEAPS algorithm is the basis of the inference engine. LEAPS. The facts are class objects; in the case of chatbot they are the fragments of sentence written by the user and included in the WorkingMemory class. WorkingMemory is the most important class of rule engine. It is used to store the references to the objects (facts) and to enable the access to communication interface with the external program.

The entry of a new fact into the Working Memory consists in its check in respect of rules matching. After passing the assertions, the handle representing the fact in the WorkingMemory and enabling the operations performed on that fact, is returned.

3. Rule structure and attributes

The rule in the framework of JbossRules is defined as the structure consisting of two parts i.e. Left Hand Side (LHS) and Right Hand Side (RHS) as in the case of Jsr94 engine originating from JESS. LHS element consists of conditional elements. When the facts are entered into the Working Memory, the trial of their match to LHS elements is made. In the case of the fact matching to the LHS phrase, the operations being performed are recorded in RHS.

```
Example:/*50% discount for each Client*/
rule "50% in USD"
when
   k : Client()
then
   System.out.println("Super promotion – discount 50%");
   k.setDiscount(0.5);
end
```

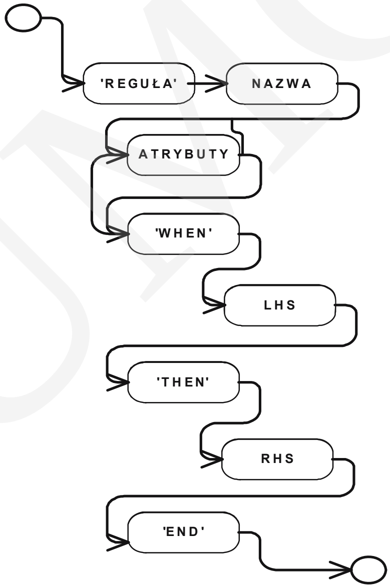

Fig. 2. Algorithm of Left Hand Side (LHS) and Right Hand Side (RHS) rule application in JBossRules

Agenda using the problems resolving strategies is responsible for the establishing of activation sequence for the rules and their association with the facts.

The function associated with conflicts resolving is similar to the case of other types of such solutions (e.g. JESS). When several rules are met, i.e. there are several rules awaiting „kick-off" in the queue, the most optimal sequence must be selected by the engine („kick-off" of rule A may result in the removal of rule B from the queue). The following options are used for conflicts resolving: Saliance – with the priority of the rule to be established by the user (larger number for larger priority), LIFO (last in first out) – based on the counter used in Working Memory Action.

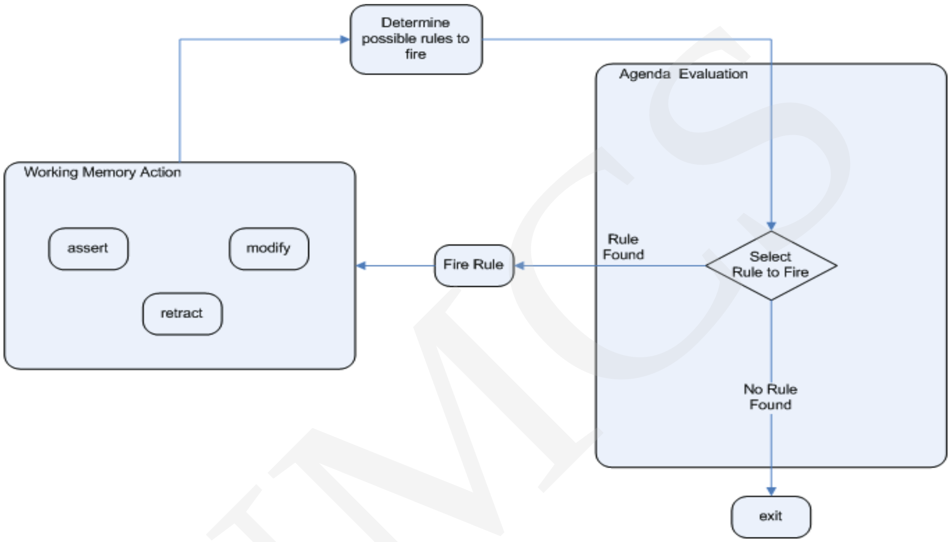

Fig. 3. Algorithm for rules „kick-off" in JBossRules

4. Example of conversation with chatbot and simple method consisting in sentence context determination

Student: I do not understand what exactly is binary addition, could you help me, please?

All fragments of the sentence, from the individual words up to the complete sentence, are entered into the rule engine as the facts and checked for any potential rules existing for them. After checking, the rules for the following fragment have been found:

1) „I do not understand"
2) „what exactly is"
3) „binary addition"
4) „could you help me, please"

The following sentence is the reply to the question asked by the student:

Chatbot: OK., I will try.

The task of the rule in a/m fragment is to find the key words on „its right side" in order to understand the idea of the question. Having applied several operations, chatbot found the text „binary addition" included in the rules group being the answers to the questions asked by the user. Then corresponding information on the binary addition is displayed and the „binary addition" is entered into global variable Context as the subject of conversation in order to focus the attention of chatbot on the presented problem in the next sentences of the user.

In a hypothetical case, any question idea will not be found in the facts base, the whole problem is not understandable for chatbot using so called filling sentences (in the scope of a few or more than ten categories) or asking the user to express His/ Her question in a different form (as the approach being frequently used by the teachers) e.g. „Could you ask your question in a different manner, because I am not able to understand it? I am not Einstein". The question is asked again by the user and analysed in an analogical way and the new rules which can be „the idea" of the question are added to the group of rules from the preceding sentence.

Summary

The choice of technology is one of the most essential decisions to be made before any attempt to implement a chatbot (except for its functions to be performed). Trying to make the best choice from among the options available in that market sector, we decided to combine the JbossRule rule engine with the entity components. The latter have been applied by us in combination with Java Server Faces technology in the components of our tutorial system, i.e. Interface AuthorInterfaces, Quiz, Intelligent Tutorial System. We hope to achieve conflict free integration of system elements in future owing to the modular structure of the programmed elements.

References

[1] Rutkowski L., *Metody i techniki sztucznej inteligencji*. Wydawnictwo Naukowe PWN, Warszawa, (2006) 435, in Polish.
[2] Byung-In Cho, *Dynamic planning models to support curriculum planning and Multiple Tutoring Protocols In Intelligent Tutoring System*. Illinois Institute of Technology, Chicago, Illinois, July (2000) 112.
[3] Zhou Yujian, Evens Martha W., *A Practical Student Model in an Intelligent Tutoring System*. Proceedings of the 11th IEEE International Conference on Tools with Artificial Intelligenc., Chicago, IL, (1999) 13.
[4] Graesser C., Lu S., Jackson G.T., Mitchell H., Ventura M., Olney A., Louwerse M.M., *AutoTutor: A Tutor with Dialogue in Natural Language*. Behavioral Research Methods, Instruments, and Computers, 36 (2004) 180.
[5] Rickel Jeff, Lesh Neal, Rich Charles, Sidner Candace L, Gertner Abigail, *Collaborative Discourse Theory as a Foundation for Tutorial Dialogue*. In Proc. Sixth International Conference on Intelligent Tutoring Systems, Springer-Verlag, June (2002).
[6] Alpert Sherman R., Singley Mark K., Carroll John M., *Multiple Instructional Agents in an Intelligent Tutoring System*. International Workshop on Instructional Uses of Animated and Personified Agents (at the 9th Int'l Conference on AI in Education), (1999).
[7] Gocłowska B., Łojewski Z., *Intelligent Tutorial System LISE*. Annales UMCS, Informatica AI, 5 (2006) 441.

[8] Russell, Rory Scott, *Language use*. *Personality and True Conversational Interfaces* http://www.geocities.com/rorysr2002/
[9] Abou-Jaoude S., Frasson C., Charra O., Troncy R., *On The Application Of A Believable Layer In IST*. AI-ED ’99 Workshop on Animated and Personified Pedagogical Agents, (1999).
[10] Wallach R.S., *Notes on the Loebner Grand Prize Rules* http://www.alicebot.org/articles/wallace/grand.html
[11] Friedman-Hill Ernest J., Version 7.0p1 (21 December 2006) DRAFT, Sandia National Laboratories, Jess®, the Rule Engine for the Java™ Platform
[12] Giarratoano J., Riley G., *Expert Systems Principles and Programming*. Thomson Course Technology, (2005).