An Intelligent Dialogue For Online Rule Based Expert Systems

Sascha Mertens
Institute for Multimedia and Software Engineering
Oststr. 99
47057 Duisburg, Germany
+49 203 379 1156
sascha@uni-duisburg.de

Marius Rosu
Institute for Multimedia and Software Engineering
Oststr. 99
47057 Duisburg, Germany
rosu@fb9dv.uni-duisburg.de

Yuliadi Erdani
Institute for Multimedia and Software Engineering
Oststr. 99
47057 Duisburg, Germany
+49 203 379 1074
yul_erdani@uni-duisburg.de

ABSTRACT

This paper describes a concept for creating free configurable, intelligent behaving web dialogues for rule based expert systems. Free configurable is meant to indicate, that the dialogue module developed with this concept is domain independent and being configurable without needing means of programming. Intelligent means that it in spite of this independency, it can behave in accordance to the expert system's knowledge and the received user inputs.

Categories and Subject Descriptors

I.2.m: [**Computing Methodologies**], Artificial Intelligence, *Miscellanous*

General Terms

Performance, Design,, Human Factors, Standardization

Keywords

Expert Systems, Rule Based Systems, Intelligent Dialogue, Dynamic Web Interface

INTRODUCTION

Rule based expert systems are widely used nowadays, mainly for tasks like diagnosis and prediction. As a typical scenario there is a set of observable facts (e.g. symptoms) that are given to the system which uses its rulebase on this observed facts to deduce a final statement i.e. a diagnosis, a prediction or a decision. During our development of a rule based online student consultation system we had to realise, that the intelligence of the whole system is limited by the intelligence of the interview it conducts with the user. On the following pages we try to describe the techniques we involved to provide such an intelligent interface.

BACKGROUND

The concept described in this paper is a consequence of the requirements of a project on a web based student consultation system. The purpose of this expert system is to offer interactive consultation to students from all over the world who are interested to start studying at our University. One main focus of this consultation is to give the students a prediction on their chances on admission as well as sketching out a hypothetical, individualised study plan. Therefore the system is supposed to interview the interested students about their educational background (primary & secondary education, degrees, language skills etc). The gathered facts are evaluated with a rule based reasoning engine that generates a prediction on the student's changes on getting admission, and proposes a study plan fitted to the student's background.

Many common expert systems (e.g. [2]) have a restricted and small set of observable facts that does not change very often. In such a case, user interface design is quite straightforward as there is no need to make it configurable nor having to employ complex mechanism to control the interview. In contrast, the described consultation process and the respective interview highly depend on how much and which knowledge is available on the educational system of the actual user – if grading system, degrees, universities and their curricula are known for a certain case the interview can be precise and comfortable; for students from 'unknown' countries a more unspecific interview must be used. That means that the set of observable facts for the student consultation is

- extensive
- different for each user
- changing, when new knowledge is added

Therefore the first main requirement for the dialogue is that the system reacts flexible to the users input, leading the most appropriate and comfortable dialogue that is possible with the given knowledge. The second main requirement is easy configurability so that the dialogue can be extended when new background knowledge is gathered. This means that elements and flow of the interview must be manageable without any re- or new programming. Both requirements imply that in its core, the interview must be conducted by a dynamically created dialogue. In the following we describe a 3-step concept for a web based dialogue that addresses these requirements.

THE GENERIC WEB DIALOGUE

In a first step we wanted to sketch out a generic web dialogue. Following the principles of object-oriented analysis, the aim was to design a class model of a web based I/O module that is flexible enough to request all kind of information needed by our system. The smallest "unit" or basic class of the generic dialogue is what we call a subquestion: a leading text, an input field and a following text. As web technology offers free inputs as well as selection we had to decide what to use and what to exclude. Free input has the advantage of giving the user a bit more of the impression of being able to input whatever s/he likes as well as interacting with some intelligent entity. On the other hand, the underlying expert system would have to extract predefined information from the free

IUI'04, Jan. 13–16, 2004, Madeira, Funchal, Portugal.
ACM 1-58113-815-6/04/0001.

Figure 1: Screenshot of a question page

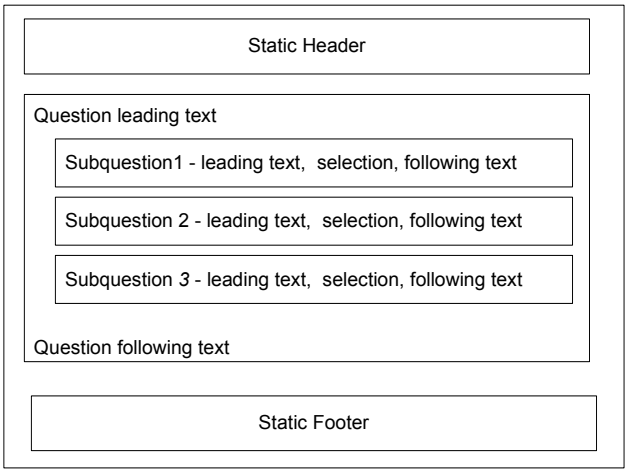

Figure 2: The layout for the question in Figure 1

input anyway, thus free input comes along with information that cannot be interpreted (like wrongly typed numbers or those out of range) or is simply not used by the system. Therefore free input unnecessarily prolongs the interview and it seems sensible to employ a dialogue that only makes use of selections. Thus our subquestion class consists of a leading text, a number of answers to choose and a following text. In HTML this is a form with a piece of text, one of the selection types radiobutton, checkbox or selectbox and another piece of text. The choice of the user defines one fact, thus the class subquestion needs to contain the information about which fact it is bound to. The basic idea here is, that for every fact used in the expert system there is one instance of a subquestion.

Considering that receiving and sending internet pages is time consuming it would be a waste of time and bandwidth to built one page for each of this subquestions. Moreover, in many subscenarios of our consultation systems the subquestions are heavily related to each other (e.g. when asking for the courses of one semester) so it makes sense to ask these questions on one webpage. As a consequence, subquestions are grouped together to form the "question" class, consisting of an introductory text, *n* subquestions and a following text. To complete the web page we added static headers and footers, containing non-changing parts as titles, logos, menus, contact info etc. Figures 1 and 2 show the example of a page containing one question that consists of three subquestions (radiobuttons, selectbox, radiobuttons).

THE DYNAMIC WEB DIALOGUE

The second step is to use the described concept of a generic dialogue to dynamically built the according webpages. For this purpose instances of the dialogue's classes are needed. The attributes for these instances are stored in a database as shown in Figure 3. A server-sided script (display script) reads the data that defines the actual question and dynamically creates the according webpage. This page contains an HTML form where all form elements have names that were automatically generated by the script. In this way a second script (evaluation script) can read the users input after the form was submit. The inputs of each subquestion can than be assigned to the corresponding fact.

The database approach ensures the required configurability: to create or modify dialogue elements we only have to modify database entries. Thus we can simply write a database interface that allows non-programmers to configure the dialogue elements without bothering them with HTML, database theory or knowledge engineering background. Still this approach is completely independent of the knowledge domain – and totally non-intelligent.

THE INTELLIGENT DIALOGUE

As a third step we wanted to use the dynamic dialogue module to conduct an intelligent interview. For a dialogue based on questions with sets of possible answers, an intelligent interview is manifested in

- a smart choice of the next question.
- sets of possible answers that reflect the system's knowledge.

Thus the concept for an intelligent dialogue has to ensure, that acquired knowledge is not only used for the reasoning itself but as much as possible for the interviewing as well. Therefore the choice of the next questions is a task for the knowledge based part of the system. This simply means that the knowledge base is split in two parts: one part containing rules to decide which answers to give and one part to decide which actions to perform next. The according system – user interaction is provided by a structure as shown in Figure 4 and has the following sequence:

- The display script delivers the actual questions as a HTML form to the student.
- The user answers the questions by filling the form.
- The evaluation script looks up the dialogue database for the

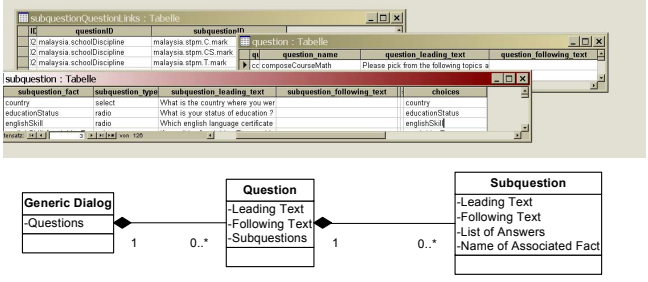

Figure 3: Instances of the dialogue, stored in a database

names of the facts that were asked for. It reads the form and adds the newly found facts to the system's runtime knowledge.

- The knowledge engine is started and determines the next action. This information is delivered to a navigation module as a fact.
- If the next action is asking more questions, the display script receives the questions' IDs as facts and composes the according pages using the dialogue database.

Using this technique, the behaviour of the user dialogue is fully determined by the rules in the rule base and can be configured freely without the need of any programming.

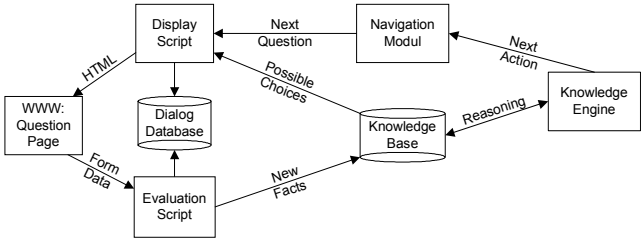

Figure 4: The basic structure of the consultation system

The problem of presenting an intelligent set of possible answers is again solved with an interaction between dialogue module and the knowledge part of the system. The idea is that if we present an individualised list of possible answers to the user we require to have *knowledge* on something: asking somebody for his degree's grade and presenting as possible answers the list of grades as being used in his home country implies having knowledge on the grading system of this country. It is therefore sensible to store this information in the knowledge base and not in the dialogue. Accordingly we have set up a generic topology to represent education systems. For each subquestion the dialogue database contains an implicit reference to the needed knowledge. Implicit means that the reference is first resolved by using the facts already entered by the user before the knowledge base will be consulted for the needed list of answers. In this way the combination of rules and input facts determine, which questions to ask and the combination of input facts and previously known facts determine the actual appearance of each question. With this technique the *design* of the dialogue remains independent from the application domain, which is one of the main obstacles in expert system interface development ([3]).

CONCLUSION

This paper describes a concept for designing a fully configurable, intelligent acting interview model for rule based expert systems. This concept is independent of the actual domain of knowledge. As dialogue and reasoning engine *only* communicate by exchanging facts, it can be used with any expert shell or reasoning tool that can be adapted to provide simple knowledge interchange functions. The dialogue itself can be realised in any language that supports server-sided dynamic creation of HTML pages.

This concept has been successfully implemented within a knowledge based online student consultation system, realised with Active Server Pages, a mySQL database and a self-written reasoning engine. Being part of this system, the interview dialogue has proved its power to be quickly configurable to provide any needed dialogue elements. The actual quality of the dialogue naturally depends on the amount and quality of the knowledge – the interview can only be as intelligent as the facts and rules that control it.

Future work on the intelligent dialogue concept will mainly focus on the attempt to increase the connection between the interview's proceeding to the knowledge topology. By linking every question to its corresponding fact and maintaining a strict hierarchy for previously known facts as well as observed facts we think that it is possible to create a self-organizing dialogue. A second focus will be on the possibility of providing the dialogue itself with more functionality like type checks or other client side interactions.

REFERENCES

[1] Carlos-Andres Pena-Reyes, Rosa Villa, Luis Prieto, Eduardo Sanchez: COBRA: An Evolved Online Tool for Mammography Interpretation, *http://lslwww.epfl.ch/~penha/docs/iwann03_cobra.pdf,* Sept. 2003

[2] "Grad Admissions Advisor", *Acquired Intelligence Inc.*, 1996, http://www.aiinc.ca/demos/grad.html, Sept. 2003

[3] V.V. Gribova, A,S., Kleshchev: Model of the Interface for Managing Verbally Represented Source Data in Expert Systems*, Journal of Computer Systems Sciences International*, Vol. 40, N^{o} 1, 2001, pp125 – 130.