Enhanced User Interfaces for Diagnostic Expert Systems

Yasser Abdelhamid

Institute of Statistical Studies & Research – Cairo University – Egypt

Yasser@infinity.com.eg

Abstract

The acceptance of a knowledge-based system by the end user has been regarded as one of the major criteria for its success. The user interface design is one of the main reasons that hindered knowledge-based systems transition from prototype to everyday use. The success of knowledge-based systems therefore depends on the user interfaces as well as the efficiency of its knowledge encoding and reasoning.

In this paper, the problems encountered in the currently applied user interfaces of knowledge-based systems are discussed, and a number of enhancements are introduced for designing diagnostic expert system user interfaces.

1. Introduction

User interface is the most important constituent of any software application especially for knowledge-based applications because of its heavy interaction with the user [1] [2]. Among these applications, come the diagnostic expert systems that require lots of information from the user to reach a perfect solution. Many approaches have been introduced for providing satisfactory user interfaces, but still, they have problems to be solved.

In this paper, we shall describe the different approaches for developing user interfaces for diagnostic expert systems, and the problems encountered by each of them, and then we shall introduce the design and implementation of our proposed approach.

The cattle & buffalo diagnosis expert system is a project that was developed by the Central Laboratory for Agricultural Expert Systems (CLAES) in collaboration with General Organization for Veterinary Services (GOVS) in Egypt [3]. The user interface of this system has been modified using the proposed approach, and the results were satisfactory.

2. User Interface Development Approaches

There have been a number of approaches for developing user interfaces for the diagnostic expert systems. In this section, we shall describe each of these approaches along with the problems encountered by each of them.

2.1. Dialog-Based Approach

The classical approach of user interaction in expert systems is based on building a dialog between the user and the expert system [4]. This dialog is a sequence of questions presented to the user one by one, and according to the answer provided by the user to every one of these questions, the next question is generated. The order of presented questions is also determined according to the order of the contents of the rules in the knowledge-base, and on the sequence of inferences in the task structure.

This approach has been adopted by most of the classical knowledge-based systems, but it has the following problems:

- According to this approach, the user goes through very long sessions, during which he/she is obliged to answer a lot of questions; some of them can be omitted if the questions are presented in a different sequence.
- The user has to answer all the questions in order to get the reasoning results. But sometimes the user does not have the answers to all the presented questions, and needs to reach a solution according to the answers he has already given to the system.
- Sometimes the questions are not presented to the user in a logical sequence. This is due to the fact that questions are generated according to the order of rules in the knowledge-base, and the order of premises and actions that constitute the rules.
- Sometimes the user needs to know why some questions are presented. The ‘WHY’ explanation that gives the state of reasoning does not always satisfy the user.
The user does not have the facility of choosing the questions that he has answers for; he is obliged to follow a sequence of questions presented by the system.

The user can not go back and forth through these questions, so he can change the answers that he has given in previous questions.

This approach does not allow the user to view the reasoning progress, and the effect of his/her answers during the session. Only the final results are presented to the user at the end of the session.

2.2. Fixed Forms Approach

This approach depends on grouping the questions to be presented to the user into a number of forms. According to the answers provided by the user, another form of questions is presented.

This approach has solved some of the problems of dialog based approach, like going back and forth from one form to another, and collecting questions in a logical sequence, in addition to reducing the number of interactions with the user. But still some problems not solved, in addition to new problems specific to this approach regarding the number of questions presented in one form, which sometimes become too many, that makes the user misses some questions, the matter that affects the reasoning process.

2.3. Dynamic Forms Approach

According to this approach, only one form is presented to the user, initially with all the possible questions that he/she might have answers for. The user is allowed to pick-up any question or property to give an answer or value. So the sequence of answering the questions is no more the responsibility of the system.

As the user gives values to the available properties, the system dynamically filters the irrelevant properties, so the rest of properties available to the user are just the relevant to the values that he has already given to the system. This reduces the number of questions that the user may answer to the minimum. The user is also allowed to retract, and give other values to some properties.

This approach has solved most of the problems described before, and has been applied to the final product of the cattle & buffalo diagnosis expert system [4]. Although this approach was very much acceptable by the users, it has the following drawbacks:

- The user has to search within a very long list of properties to find the one he needs to fill in its value.
- Although the user has more freedom in directing the reasoning process, he is only allowed to do this through altering the entered data.

3. Design of the proposed approach

The cattle & buffalo health-care management expert system [3] is a project that was developed by the Central Laboratory for Agricultural Expert Systems (CLAES) in collaboration with the General Organization for Veterinary Services (GOVS). The developed system is mainly a diagnosis expert system that covers all disorders affecting cattle and buffalo species.

The user interface of the prototype model of the system was based on fixed forms approach, where the properties were partitioned into a number of forms, and the user navigates through these forms. This prototype was not successful, and was rejected by the end users because of the problems discussed before.

The final product of the system adopted dynamic forms approach, and it has solved most of the problems as mentioned before. Figure [1] is a snapshot of the final products user interface.

![Cattle & buffalo diagnosis expert system user interface](image)

The proposed design presented in this paper offers two improvements. First: the user is allowed to enter his findings through a graphical navigation tool. We believe this will allow the user to move directly to the required property, without having to search within long lists of choices. Second: the user is allowed to direct the line of reasoning of the expert system, not only by
providing case data, but also by selecting the inferences to be applied at any time during the session.

3.1. The graphical navigation tool

The successful user interface is the one that provides the user with the tools that help him in locating the required information at the shortest time, in addition to being consistent in its behavior and being logical in the sequence of actions upon each move taken by the user.

The basic idea of our work is to introduce a graphical tool through which the user can navigate the different organs of the animal, and select the one that he has findings for.

In the implemented prototype, we used a two-dimensional figure of cattle. The organs/regions of the animal are represented graphically as polygons. As shown in figure [2] each organ/region has a name and an area on the figure. The shaded area represents the location of the marked region.

The internal representation of these polygons is shown in figure [3] where each region is represented by a name and a list of data values that define its area and location on the figure.

![Figure [2] Regions defined by the graphical navigation tool](image)

![Figure [3] Internal representation of organs/regions polygons](image)

The knowledge base of the original system has been modified to include the association between organs and related properties. The user can navigate through the figure and point at the desired region. When an organ or an area of the figure is clicked, the properties related to this area are displayed to the user so he can select the desired property. When the user selects another area, only the properties that conform to the previously entered findings are displayed.

To exhibit the effect of using the navigation tool we have added a control to show the number of items in the list of properties. As we can see in figure [4-a] the number of selectable properties is “95”, while in figure [4-b] the selectable properties are only “10” upon selecting the abdomen using the graphical navigation tool.

We can notice that only the properties relevant to the selected region are displayed in the list of properties. Also, we can notice that only the properties that conform to the previously selected findings are displayed according to the available knowledge.

3.2. User directed reasoning

In traditional expert systems the session is guided by a predefined task structure. This task controls the behavior of the knowledge-based system from the start to the end. Naturally, there is more than one reasoning model for each problem type [2], but only one is selected by the knowledge engineer according to the characteristics of the application at hand. There is no flexibility in selecting the line of reasoning at run-time.
Our proposed idea is to provide the user with the capability of switching to different lines of reasoning at run-time. This can be done by providing a number of inferences, and providing the facility of selecting from at run-time.

The following is a list of inferences that the user can use during the session:

1. **Change mode of inference:** The user can alter the way the system includes suspected disorders. There are two modes, the first does not exclude disorders that has properties other than the properties selected by the user. For example if the user has selected a property with a certain value, and there is a disorder that does not have this property within its knowledge, then this disorder will not be excluded from the suspected disorders. This mode conforms to multiple fault assumption, which assumes that the case may have more than one disorder. The second mode includes only disorders with properties having the same values entered in the case data. This mode conforms to single fault assumption, which assumes that the case has only one disorder.

2. **Discard a disorder:** Sometimes, during the phases of the session, the system suspects more than one disorder, but the user can discard one or more of these disorders to focus on the real suspected disorders.

3. **Excluding a value of a property:** The user can exclude a value for a certain property, like negating its existence.

4. **Suggesting a disorder:** Sometimes the user suspects a certain disorder even before he runs the diagnostic expert system. In this case, the system can help the user in confirming this suggestion by displaying only the properties related to this disorder. If the user could enter the case data according to the available properties, then the disorder is confirmed, otherwise it is not confirmed.

5. **Differentiate:** Sometimes the user reaches a point where multiple disorders are still suspected. The user can ask the system to help him in generating a detailed differentiation table for these disorders.

6. **Display typical findings:** The user can display the typical findings for one of the suspected disorders.

**4. Conclusion**

Diagnostic expert systems are characterized by its heavy interaction with the user. The user interface of such systems is the primary factor for its acceptance by the end user. In this paper we have introduced two major enhancements for the diagnostic expert system user interfaces.

The first enhancement was a graphical navigation tool that helps the user to go directly to the required property without having to search in long lists. The second introduced enhancement is allowing the user to alter the line of reasoning by providing a library of inferences that the user can call at run-time.

From the simple case presented hereafter, it seems to be a promising idea especially when sophisticated graphical tools, and multimedia techniques are applied. For example, the animal can be modeled as a three-
dimensional object instead of a two-dimensional graphical object. This will give more realistic feeling to the user. Also virtual reality techniques can be applied for this reason.

Further enhancements can be added by augmenting multimedia capabilities to model the behavior of the model in the form of sound and animation.

The second achievement of this work is shifting diagnostic expert systems from the state where all the control is coded in the task structure of the expert system, and the role of the user is just providing the required case, to the state where the session control is shared between the user and the expert system, giving the opportunity to the user to alter the line of reasoning by means of a number of inferences that he/she can call at run-time.

It is expected that the developed graphical navigation tool will succeed in diagnostic expert systems in other domains like plants and human beings.

Finally, the implemented graphical navigation tool is expected to be very helpful in knowledge acquisition, where the domain expert is allowed to attach properties and property values to the predefined regions of the graphical object.

5. References