CONGAXPERT – AN ONTOLOGY BASED ONLINE EXPERT SYSTEM

Rosu Marius, Phan Cong Chinh, Stefan Freinatis

Institute for Multimedia and Software Engineering
Oststr.99, 47057 Duisburg, Germany
{rosu, chinhha, freinat}@uni-duisburg.de

Abstract: Today's expert systems deal with domains of narrow specialization and the set of observable facts is limited. When trying to enlarge the specialization domain, the limited number of facts becomes inaccurate. This paper presents an attempt to solve the inaccuracy problem of wide domain expert system by modeling the domain of knowledge and formalizing the reasoning process. The intelligence of the systems is used for selection of possible observable facts as well as for the prediction or diagnostics function of the expert system.

Keywords: rule based expert system, ontology, intelligent dialogue.

1. INTRODUCTION

Since their appearance in the 1970's, rule based expert systems have found applications in a variety of domains ranging from control systems for power plants to help systems. They are mainly used for tasks like diagnostics or predictions. The typical scenario consists of interviewing the user in order to acquire facts that are used afterwards to generate the predictions or diagnostics. Many common expert systems have a reduced number of observable facts and the intelligence of the system is used only for the process of generating the prediction. Most of them do not take into consideration the importance of this set of observable facts. The limited number of observable facts induces a generality problem that affects the quality of the observable facts. In the author's opinion, the quality of a decision is highly dependent on the quality of the observable facts. But by increasing the level of specificity of the observable facts, the quantity of facts is considerable increased. This would induce the need of using special selection methods for these facts that will reflect in the use of intelligence in the conducted interview. This paper will present a concept of an intelligent system that efficiently combines the intelligence of the conducted interview with the intelligence of the rule based application used for issuing predictions.

1.1 Background.

In 1997 the Faculty of Engineering Sciences at the University Duisburg-Essen started the internationalization of its study programs. Since 2001 the faculty in particular runs the Bachelor/Master program International Studies in Engineering (ISE). For the advisory service and for the administration of the ISE program, a set of software tools has been developed at the faculty. One of these tools is CongaXpert project - an online consultation system intended to offer interactive consultation for students all over the world, who want to join the program studies at University Duisburg Essen. The main purpose of this consultation process is to give users a prediction on their chances to be accepted. An interview is carried out in order to obtain information referring to the education background (primary & secondary education, degrees, language skills etc.) of the interested students, as well as their interest for a particular study program at our University. The gathered facts are evaluated with a self-written inference engine that generates a prediction on the admission and proposes a fit study plan based on user's background.

1.2 Observations

Many common expert systems (e.g. Pena-Reyes et all.) have a restricted set of observable facts. In this case, the development of the user interface is quite straightforward, since there is no need for special configurability methods or for complex mechanisms to control the interview. This is not the case for the previously described system. The quality of a decision highly depends on the quantity and quality of the available knowledge referring to the educational system of the present user. If grading systems, degrees, universities and their curricula are known, the interview is precise and efficient. If no knowledge referring to the educational background of the user is available, a much more general and imprecise interview must be used. Other elements that are worth taking into consideration are the dynamics of the education systems all over the world as well as the structural differences between them. This implies that the set of observable facts for the student consultation is:

- extensive;
- different for each user;
- evolving, when new knowledge is added, or when education systems are modified.

The main requirement for such a system is, therefore, flexibility. The system must react to user input in order to conduct an interview in the most efficient manner. Another requirement is the configurability of the dialogue such that the evolved knowledge can be appropriately expressed by the interface. This means that the contents and flow of the interview must be manageable without need of reprogramming.

Therefore it makes sense to separate between the intelligence used for issuing the prediction concerning the student's chances to get accepted in our university and the intelligence of the conducted interview. By this approach we can ensure an intelligent reaction of the system to the evolution of knowledge without interfering with the reasoning process of the inference engine. An intelligent user interface increases the quality of the observable facts by providing means of selection for the relevant facts.

2. THE KNOWLEDGE BASE

The main challenge during the development of this project was structuring the domain of knowledge in such a way to ensure the required flexibility. The extensive character of this knowledge required a dynamic way to address the knowledge. The fact that the knowledge is different for each user imposed the need for a way to address the knowledge in a general or conceptualized way. The answer to these problems is represented by the ontological knowledge bases. Therefore an ontological knowledge base has been developed. From particular elements of knowledge, there have been inferred concepts or classes used to reflect the structure of knowledge in the knowledge base. These concepts provide a way to address the particular elements of knowledge (instances of knowledge) in a general way. From this moment we shall use "class" to denote the concepts and "instance" to denote the particular elements of knowledge.

2.1 The world graph of knowledge

Modeling the knowledge base was particularly difficult because of the high heterogeneity of the education systems all over the world. The inferred concepts had to reflect in the most accurate manner possible the structure of all the education environments. The used formalism contains "classes", "properties", "instances" and "relationships".

- "Classes" represent concepts inferred from the domain knowledge based on classification. They are expressed as a collection of attributes of type class or property Ex: "Country", "University", "School";
- "Properties" are the primitive data types and represent conceptualizations of sets of values. For example "Mark" represents a concept derived from simple values like {1, 2, 3,..10};
- "Instances" represent the actual knowledge from the knowledge base. Ex :"Germany", "University Duisburg-Essen", "default School Germany";
- "Relationships" represent binary relations between classes. They can be classified as interconcept relationship such that the one of inclusion, in the case of classes having attributes of type class, and interconcept relationship – equivalence between different instances of the same concept

The relationships play a critical role in this framework. They are the ones that ensure the intelligence of the interviewing process by providing the necessary means for selection. One of the most important relations is the inclusion relation between concepts. This relation defines the ways in which the concepts are actually built using other concepts as attributes. Hence the structure of the knowledge base is the one of a directed acyclic graph having higher level concepts connected to their attributes through inclusion relationships.

The way of addressing a particular instance or a group of instances is done through these relationships. When identifying the instance or the group of instances, the system needs the concept that has been defined for this instance as well as instances related with the target instance.

One significant case is the one when "University" and "University Degree" are sharing the same property "mark" but at instance level the marking systems applicable for "University" or university courses are different than the one used for "University Degree".

In Fig. 1 is partially presented “The world graph of knowledge” representing our system’s ontology. The classes or concepts are starting with capital letter, whereas the properties are names starting with lower case.

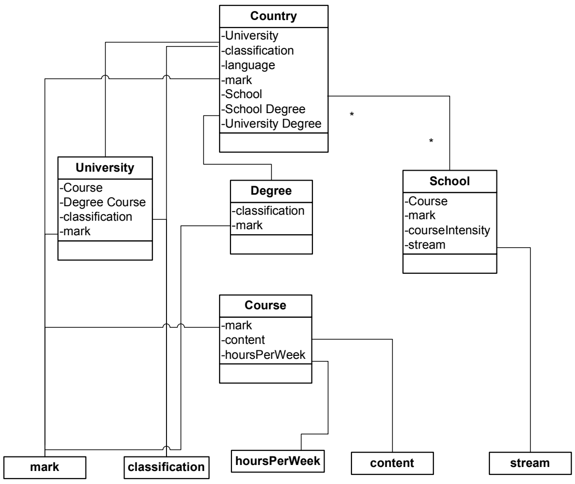

Fig. 1: The world graph of knowledge

The instance level adds a third dimension to this drawing through concept instances, relationship instances or property values. There may also be the case when for a certain property there are no values as attributes of particular instances. In this case, default values have been provided.

2.2 *The Student Tree*

The generality achieved by formalizing the access to our knowledge base is not sufficient for reasoning. The typical scenario supposes that the facts from the pool of facts represented by the world graph of knowledge are selected, displayed to the user and the user makes a further selection. In many cases it appears the situation that the formalism of the ontology language to be incoherent to the expert system rules. This comes from the contradiction between the generality intended by the Ontological Knowledge Base and the particularity of a reasoning engine. In this case a different formalism is necessary in the fact representation for the inference engine. This allows us to detach from the „shared conceptualization”(Towle et all.) presumed by an ontology and create an own formalism necessary for the reasoning engine. The most relevant example is the one with the chicken and the egg. When we want to use egg ontology, the chicken comes out of the egg, but if our intention is to find out how many eggs a chicken lays, the relation is reversed and a formalism related to the chicken is needed. Moreover, both chicken and turkey lay eggs. But we are interested which eggs are from the turkey and which are from chicken. In this case, egg concept can no longer be shared between chicken and turkey. This implies that the knowledge representation model is no longer a graph, but a tree, containing chicken eggs and turkey eggs specialized concepts instead of the general egg concept.

When applying the above mentioned principles to the case of education systems, we obtain student formalism that we called “The Student Tree”. This tree contains the information that is needed for giving a prediction of the student’s chances to be accepted to our university. It contains at the beginning of the consultation process only concepts or classes, following that during the interview this tree will be filled in with instances.

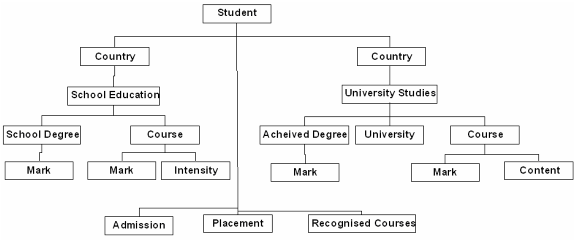

Fig. 2: The student tree

3. THE REASONING ENGINE

The fact formalism described in the previous section imposes similar formalism for rules. The facts that are being fed to the inference engine have a formalized structure as shown in figure 3. The structure of a fact is instance based, that is a fact is composed of a concept from the student tree, the instance chosen by the user during the interview process as well as the properties and property values of the respective instance.

Fact Object	Instance	Instance Id	Property Name	Property Value
Student.Degree.Course	Course1	Id54	Mark	5

Fig. 3: The fact format

Based on these facts, predicates are created. The predicate is the fundamental unit of the rule. A predicate is composed of a fact, an operator, and possibly a value. The operator can be a binary operator, such as the relational operators, or a unary operator such as the existence operator.

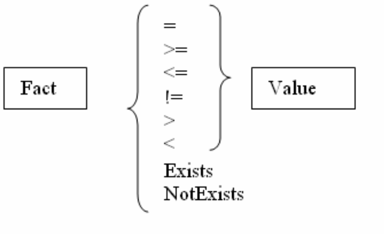

Fig. 4: The predicate format

The rules are stored in a database in order to provide the necessary configurability for the system. At the initial time, the rules contain only the concepts, following that at runtime, the facts to be received based on user inputs and the truth value of the predicates can be evaluated. From the above mentioned predicates, IF-THEN rules are constructed. The IF part of the rule contains predicates related by logic operators, the THEN part

– the conclusion of a rule and the next actions to be performed.

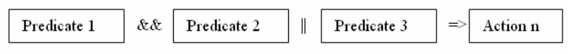

Fig. 5: The rule structure

The next action to be performed can be asking another question, creating intermediate facts and so on.

The inference engine is a self written software that can perform the rule evaluation as well as other additional tasks like rule extraction or executing actions. One important characteristic of this system is the fact that it is a web based system. This implies that at a certain moment there can be a lot of users accessing the system; hence the functionality of the inference engine has to be optimized. The rule extraction represents an optimized process such as the inference engine only loads the critical rules. This means that at a certain stage in the inference process, only the rules that contain the recently received facts are loaded. This process has double advantage, on one side it alleviates the inference engine of the unnecessary load caused by the "unimportant" rules, and on the other side keeps the logical path of the process since the next actions to be performed are strictly related to the recently inputted facts.

4. THE DIALOGUE

The typical scenario for the utilization of this system is the following: The user is displayed a "question" as a set of HTML selection elements. He is supposed to make selections and forward them to the server (answer). Here a server side script converts the answers into facts and feeds them to an expert system that evaluates them and decides what is the next question to be asked. In the next subsection it will be presented the three steps algorithm for creating an intelligent dialogue.

4.1 The generic web dialogue

In the first step we want to sketch the generic web dialogue. The smallest unit of the generic dialogue is what we called "subquestion". A subquestion consists of a leading text, an input field and a following text. For the input field we have decided not to use text fields since these ones give the user the impression that he can input whatever he liked, thus unnecessarily prolonging the consultation process. Therefore, a subquestion consists of a leading text, a number of answers to choose and a following text. Since the subquestions are highly related to each other, it makes sense to group them on the same web page. The result is the class "question" which contains one more subquestions a leading text and a following text.
To complete the page we have added static headers and footers that contain the non-changing parts of the web page.

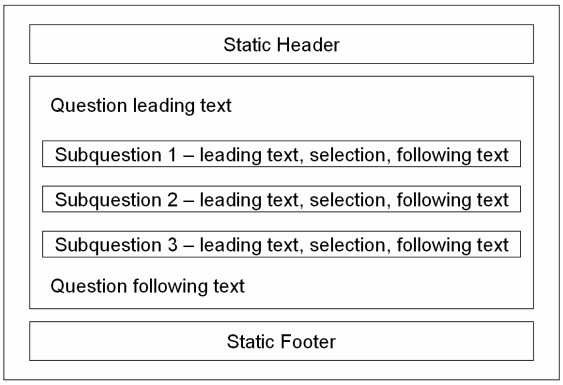
Static Header
Question leading text
Subquestion 1 – leading text, selection, following text
Subquestion 2 – leading text, selection, following text
Subquestion 3 – leading text, selection, following text
Question following text
Static Footer

Fig.6: The generic dialogue

4.2 The dynamic web dialogue

The second step is dynamically building the corresponding web pages. For this purpose, the instances of the dialogue classes are needed. A server side script creates these instances using the attributes of these classes, which are stored in a database. Every HTML element receives a unique name, which is used for identifying the values chosen by the user
The problem of displaying a set of possible answers in accordance to the system's knowledge is achieved through referencing. The subquestions are connected inside the database to the source of data or to the "world graph". The link is made at the concept level. The instantiation process is realized by receiving from the inference engine a reference to the instance of the class containing the desired instances as attributes. In this way, two of the three dimensions of the word graph are provided. The third dimension is represented by the choice of the user, thus the identification of a particular instance is achieved.
The database approach ensures the configurability. In order to modify the dialogue elements it is enough to modify the database entries.
So far, we have expressed configurability and knowledge independence of the system. What remains to be proved is the intelligence.

4.3 The intelligent dialogue

A very common mistake is confusing an intelligent user interface with an intelligent system .In order to achieve the intelligence of the interface the knowledge should not be used only for the reasoning process but for the interview as well. Because of this, we have divided the knowledge base in two parts, one that is used for deciding what action to perform and another one to decide the result of the consultation process.
Since the dialogue is based on questions with sets of possible answers, the intelligence of the system consists of:

- a smart choice of the next question
- displaying a set of possible answers in accordance to the system's knowledge

The smart choice of the next question relies on the current status of the consultation process. When new facts are added, the system is searching for missing links, or facts that are still needed for reaching a decision. This are called target facts and they are bound to subquestions. This implies that by identifying the facts that are still needed the system identifies as well the questions that should be asked next. In many cases, it is possible for the system to identify more than one question to be asked. However, the interface is designed in such a way that only one question is asked on one web page. For such situations a stack mechanism has been developed. The identified questions are put in a stack following that the interface script will take them one by one and display them to the user. In the case that the user inputs have as result another question to be asked, this is being put on top of the stack and so it will be displayed immediately after. In this way, the continuity of the interview is achieved. In order to keep the synchronization a set of flags is used.

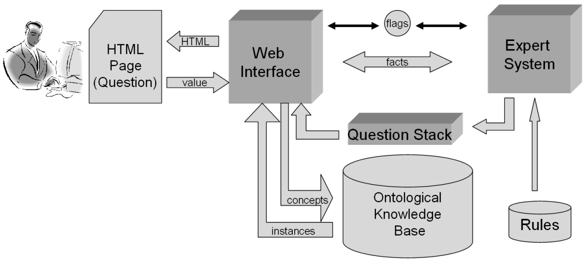

Fig.5: The basic structure of the consultation system

The problem of presenting an intelligent choice of the possible choices is solved by the way the knowledge is retrieved from the knowledge base. The interface classes are bound to the concepts thus giving the possibility to retrieve all instances representing this concept. The next step is to refine this selection by displaying only the instances significant for the user. This is done by using the history of the user selections, through identifying the instance that contains the previously selected ones as attributes. For the case when there is no knowledge to be displayed, the default knowledge is selected and presented to the user. There may be the case that for a certain concept there is only one instance available, due to the lack of knowledge or to the fact that the knowledge evolution process converted all the instances in a single, general one, because of their high similarity. In this case, the instance is automatically uploaded to the server and a new question may be displayed. In any case, the system can efficiently react to the situations that have arisen.

6. CONCLUSION

This paper describes the concept for designing a fully configurable intelligent expert system interface. The concept is independent of the domain of knowledge. Since the dialogue communicates with the inference engine through facts, it can be used with any expert system shell provided that this can be adapted to ensure simple knowledge interface functions. The dialogue can be implemented in any language that supports server side dynamic creation of HTML pages.

The configurability of the system is ensured by the database approach of the system. In order to modify an element of the web interface it is sufficient to modify the database entry.

The intelligence of the system is proved by the intelligence used in the next question to be asked and by the intelligent selection of the elements to be displayed. The knowledge evolution at instance level does not affect the interview structure. New knowledge can be easily added without being necessary to modify the interview structure. However, when new concepts are added, the addition of new dialogue classes is necessary in order to have them displayed to the user in an appropriate manner.

The navigation represents a logical sequence of events. Due to the stack mechanism, the question displayed at a certain time represents a logical consequence of the questions recently answered by the user.

The only restriction enforced by the authors is the use of text fields as a means for user input. In their opinion, this does not constitute an impediment since the entire knowledge of the system can be displayed to the user. This implies that the input that the user gives in a text field, either will be insignificant to the system, because of lack of knowledge necessary to process this input, or will be uploaded to the system in a too early stage of the interview, thus making it impossible to be evaluated.

This concept has been successfully implemented in a knowledge based consultation program realized with java and a self-written inference engine. The dialogue has proved to be easy configurable in order to provide the needs of the consultation system.

Future work will be directed towards the enhancement of the expert system's knowledge in order to increase the efficiency of the interview process. In this direction, the knowledge acquisition and knowledge evolution play a key role due to the high dependency between the quality of a decision and the quantity and quality of knowledge.

Another direction of research is towards the enhancement of the expert system's functionality by providing the possibility of issuing recommendations.

7. REFERENCES

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

Biletsky Y., Campeanu C., Dudar Z., Vorochek O. - Meta Context Mediation to Attain Semantic Interoperability. *Second IEEE International Conference on Intelligent Systems.* June 2004

Ehlert P. – Intelligent User Interfaces Introduction and Survey. *Research Report DKS03-01 / ICE01* February 2003

Frank U. - Knowledge Management Systems : Essential Requirements and Generic Design

Patterns- *ISE'2001, Las Vegas: CSREA Press* 2001 pp. 114-121

Gribova V.V., Kleshchev A,S., - 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

Mertens S., Rosu M., Erdani Y. - An Intelligent Dialogue for Rule Based Expert Systems, *IUI'04, ACM 1-58113-815-6/04/0001* pp280-282 January 2004

Middleton S., Shadbolt N., de Roure D. - Ontological User Profiling in Recommender Systems. *Http://www.ecs.soton.ac.uk/~sem99r/tois2004.html*

Pena-Reyes C. A., Villa R., Prieto L., Sanchez E. - COBRA: An Evolved Online Tool for Mammography Interpretation, Sept. 2003

Puerta A. - The Study of Models of Intelligent Interfaces - *Intelligent user Interfaces '93*

Towle B. and Quinn C. - Knowledge Based Recommender Systems Using Explicit User Models, *http://www.igec.umbc.edu/kbem/final/towle.pdf*

Waeche H., Vogele T., Visser U., Stuckenschmidt H, Schuster G., Neumann H. and Hubner S. - Ontology-Based Integration of Information - A Survey of Existing Approaches, *http://www-agki.tzi.de/buster/IJCAIwp/Finals/wache.pdf*