International Journal of Engineering and Technical Research (IJETR)
ISSN: 2321-0869, Volume-2, Issue-5, May 2014

Object representation of Patients in Diagnosis Expert Systems

DigvijayYadav, Amit Singh Gaur, JyotiBudakoti

***Abstract*— Expert Systems are generally represented as simple If-Then or If-Then-Else rules applied on a large knowledge-base. In contrast, this article tries to make an attempt to simplify the way of structuring the Expert Systems that are designed for the diagnosis purposes. We take a closer look at how the Object Oriented Design can be useful for the expert systems that are specifically designed for the diagnosis purposes. Expert systems have definitely been very hard to design and to use too, especially if the user of the system does not belong to the Computer Science background. So, this article will also try to find few ways of how the expert systems can be made easy to use or at least reduce the efforts the user has to make while using them.**

***Index Terms*— Object oriented design, Objects, Inheritance, Expert System, Abstraction, Disease, Symptoms, Features and Attributes.**

I. INTRODUCTION

Over the past two decades, the object oriented design and programming has seen tremendous use and exposure in the programming and software design and architecture too. There can be several reasons for this kind of vast use, but the most common and important reason is that now a day most of the software being designed or produced are some being related to real life instances. They intend to represent instances (or at least some portion) of the real world. Below we are highlighting some of the facilities and features that are provided by the Object Oriented Design.

A.*Objects and Classes*: Class simply represents any category of real world things and the Object is just an instance of a real world class.

B.*Inheritance*: This represents the ability of a class to inherit some or the features from another class just as seen in humans, where children inherit some of the features from their parents.

C.*Abstraction*: Object Oriented Design makes it easy to generalize things by considering only the relevant information and discarding others, for example for the treatment of toothache, problems related to eye can be ignored as it is irrelevant in the treatment of toothache.

D.*Features andAttributes*: All the objects have some features; humans have legs, some disease or some expertise, for example, and in terms of programming these features can be conceptually thought of some property of any object at any particular instance of time. This way an object is a self-contained entity which has all the data and functions that can be applied to the data and also manipulate them. Similarly, for an expert system designed for diagnosis purpose the patients can be thought of a class and all the features like physical appearance or any disease can be represented as its attribute of an object of this class on a particular time.

Manuscript received May 20, 2014.

DigvijayYadav, completed B.Tech in Computer Science and Engineering from G. B. Pant Engineering College Pauri (Garhwal) India

JyotiBudakoti,completed B.Tech in cComputer Science and Engineering from G. B. Pant Engineering College Pauri (Garhwal) India.

Amit Singh Gaur, completed B.Tech in Electronics and Communication Engineering from G. B. Pant Engineering College Pauri (Garhwal) India.

II. CURRENT REPRESENTATION

In the current scenario generally no notion exists for representing a patient as an object in expert systems. It is just a set of rules and the knowledge base that form the system. The user simply provides answers to a long series of questions and at the end the results are shown to the user. This way the user has no clue how the system is interpreting the answers and the how the results or conclusions are achieved. So including the patient as a part of the expert system i.e., representing the patient as an internal object makes the system more logical when thought from a user's perception.In the latter case, the patient is simply an object (internally) and has few attributes which are mapped to the symptoms of the real patient.

III. PROPOSED ARCHITECTURE

A. *Patient as an Object*

An object represents an instance of real world set of objects or class, similarlya patient is an instance of a larger class of living beings suffering from some kind of illness and this instance has several properties and attributes like, appearance, voice, age, gender and diseases(as he is a patient). Also since a patient was born, he can also inherit some features from his parents like color, height etc.

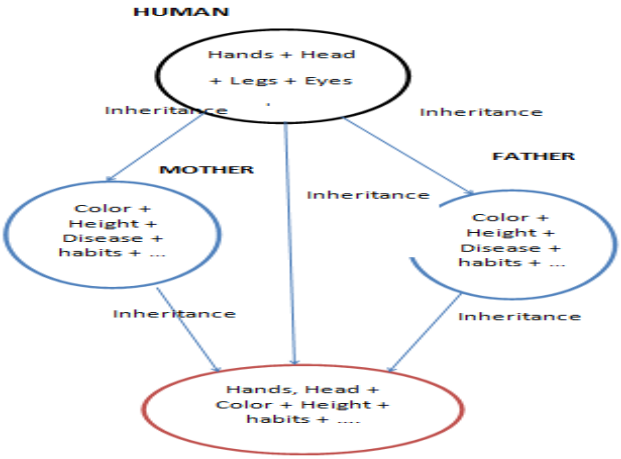

Figure 1:Class representation of patients and humans with inheritance

This way in an expert system a patient can be seen as an object with some properties and attributes some of which are inherited from the parents.

B. *Symptoms as Attributes*

An object can have several properties, for example a football has several features like, its weight, radius, and color etc., at a particular time. These attributes can get modified over the time. In the same way if a human is sick at any point of time in his life the sickness can be thought of the phase of time when the human has one more attribute which are the symptoms of the disease. These symptoms define the state of the disease and can help in deciding the cause and the cure of it. Also due to the disease some of other attributes can get affected. After the treatment the disease attribute of the patient can be removed or suppressed so it does not affect the human any more.

C. *How Inheritance can help*

Since every patient class can be derived from even more general (or higher level of) one or more classes like Men, Women, Humans and so on, there are several attributes and features that a patient class inherits from these classes. In general, these higher level classes are known as super class or parent class. Other than these, a patient inherits several features from its ancestors which again can help in finding the cause of sickness or disease, if it is some kind of inherited disease.

D. *Use of Abstraction*

When a patient comes to a physician and tells the symptoms of the disease to him, the physician thinks about the symptoms and other related issues, for example, for a simple toothache problem the physician will concentrate mainly on the food habits and habits related to cleaning the teeth of the patient and ignores other unwanted details like, checking patient's eye or ears. So essentially the physician concentrates only on relevant information and ignores other unwanted detail. This is exactly similar to the abstraction Object Oriented Design provides to the programmers.

Figure 2 is a graphical example showing/comparing disease symptom as attributes and how abstraction can be realized in this scenario with explanation of both below:

A.*Symptoms*:In the figure below, two objects are shown; **Ball object** and a **Patient.** As we can see, the features of the ball like mass, density, color etc. are the properties of the ball at a particular time and can get changed or modified with time. In the same way, if a patient is thought of representing an object then the symptoms like toothache or eye problem can be considered its internal attributes, which can also get changed over time.

B.*Abstraction:* Now to explain abstractions, consider the ball object and we have a task to find the volume of it. In this case only the mass and density of the ball are the attributes that decide the volume of the ball, hence other attributes like color or price can be simply ignored, providing us the facility to concentrate only on relevant things and ignoring unimportant things. Exactly in the same way, to find out the reason for a simple headache for a patient, we can simply ignore the leg-pain and concentrate on the other relevant issues.

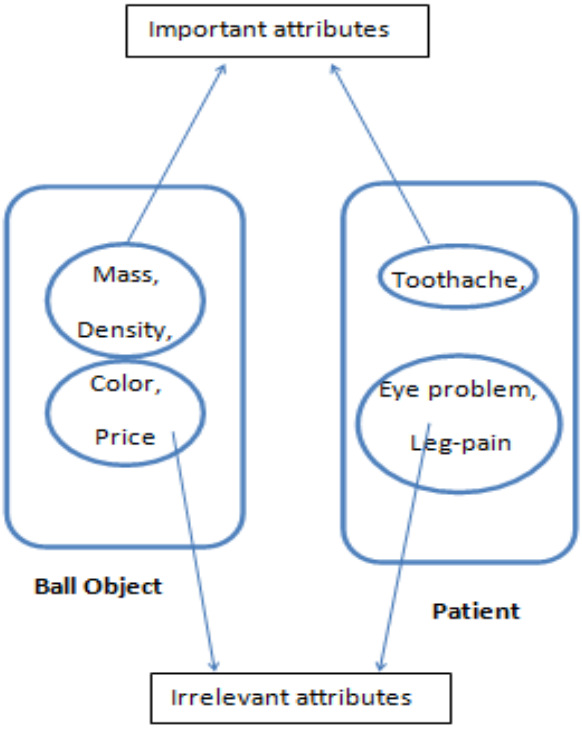

Figure 2: An example comparing the attributes with disease symptoms and demonstrating Abstraction.

IV. HOW THIS IDEA CAN HELP

Since a patient belongs to real world and the Object-oriented design helps solving problem by mapping real world objects or classes to virtual programming objects and classes, it logically can help in representing patient in virtual objects and the solving the problem in hand. This mapping can help both the designer and the user of the system as follows:

- A system designer experienced in Object Oriented Design can simply avoid the issues of traditional functional programming approach, for example designer has to deal with one object i.e., the patient and the attributes of it(which are contained in the object itself), instead of collecting data from different places and managing different data-structures to keep them safe.
- Since dealing with real world problem is fairly easy as compared to solving completely virtual problems, it is considered better to map real word object to virtual objects
- Objects can have different states at different point of time, so it can be definitely helpful to save the state of current object in the system as a file or simply be transmitted over a computer

network with the help of serialization. One of the benefits of saving the state of a patient object is that it can be referenced in future treatments. Also transmitting the saved object to other system can enable different users to share, help, and analyze the disease.

V. CONCLUSION

As we have seen, real world problems are fairly easy to solve as compared to completely virtual problems, this can make system design much simplified in contrast to the traditional functional designs. Also saving and sharing of objects can help physicians to keep track of medical history of the patient and reach more solid causes and treatments.

REFERENCES

[1] Elaine Rich and Kevin Knight *Artificial intelligence 3ED(SIE)*
[2] Dennis deChampeaux, Doug Lea, Penelope Faure "*Object Oriented System Development*"
[3] Hayes-Roth, Frederick, Donald Waterman, Douglas Lenat(1983). "*Building Expert Systems*. Addison-Wesley"
[4] Man, Marriage and Machine – Adventures in Artificial Advice, part 1http://www.codeproject.com/Articles/179375/Man-Marriage-and-Machine-Adventures-in-Artificial
[5] J.P. Hamilton "*Object-Oriented Programming with Visual Basic .NET*" O'Reilly Media

DigvijayYadav, completed B.Tech in Computer Science and Engineering from G. B. Pant Engineering College Pauri (Garhwal) India

JyotiBudakoti,completed B.Tech in cComputer Science and Engineering from G. B. Pant Engineering College Pauri (Garhwal) India.

Amit Singh Gaur, completed B.Tech in Electronics and Communication Engineering from G. B. Pant Engineering College Pauri (Garhwal) India.