PARAMENIDES: Web-Based Expert System for Problem Solving and Diagnosis

Karl Cassar
Dept. Computer Science and Artificial Intelligence
University of Malta
karlcassar@gmail.com

Abstract

An expert system is both a computer program for storing knowledge (or human expertise), and for reasoning about knowledge to come up with conclusions. Human experts have various limitations:- they can die, retire, are relatively scarce, suffer from moods, cannot always explain their reasoning, and others. An expert system has all the benefits of a computer program: easily reproduced, able to cater for a large number of users, is permanent, etc. Since human experts do not reason in discrete logic (true or false), the expert system must be able to reason under uncertainty. The Internet contains a huge repository of information that is largely unstructured. The aim of *Paramenides* is to combine the power of the Internet and expert systems into a global repository of human expertise in which knowledge is stored in a structured manner which can be used for reasoning on problem-solving and diagnosis. It is available publicly via the Internet to reach out to a global user-base, and is freely accessible to anyone.

1 Introduction and Background

1.1 Overview

The Internet can be considered one of the most important inventions of the 20th century. It has brought with it a tremendous change in how humans communicate with each other, and today we can find a great deal of information online.

However, most of this information is not structured in any manner and one needs to sift through all relevant and irrelevant information to find what he needs. This is especially true when a user uses the internet to find solutions to a certain problem. The main process involves searching on broad keywords, and constantly redefining keywords to try and close in on your subject

The idea of this project is to create a *wiki*[1] of expertise where human experts can enter knowledge in the form of rules that can be understood by a machine. Users can use its search engine to find conditions (goals, or problems) and the system can guide them to find solutions to the problem, similar to what a real-life expert would do.

[1] A *wiki* is a website that allows its users to freely manage its content.

1.2 Expert Systems

Expert Systems are computer programs whose purpose is to imitate the real-life traits of a human expert and problem-solving skills.

Jackson (1999) defines the characteristics of an expert system as:

- Able to reason about specific domain(s)
- Made up of a knowledge base, and be able to declaratively reason about it.
- It must not revolve around a set of step-by-step algorithms, but must be able to alter its reasoning according to the rules[2] in the knowledge base.
- It must be able to solve problems using an approximation method, and cannot be guaranteed to succeed

Components of an expert system

An expert system must be made up of several components, in order to serve its function:

- ***Knowledge Base*** - contains knowledge about a specific domain, as a set of rules that define the reasoning of an expert as a series of logical steps
- ***Inference Engine*** - performs reasoning on the rules defined in the knowledge bases and able to derive new facts from current known facts and rules.
- ***Working Memory*** - stores any facts inputted by the user, as well as any derived facts during the reasoning process.
- ***Explanation System*** - capable of showing the reasoning process to the user by keeping track of which rules have fired and why.

[2] A rule is one logical statement when reasoning about a problem. It contains antecedents, and consequents; Antecedents are its inputs, while consequents are its outputs.

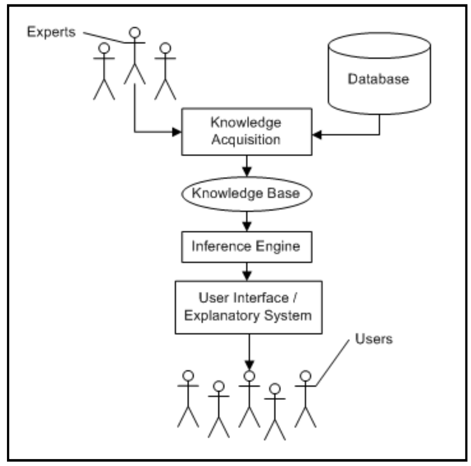

Figure 1-1: Components of an expert system

1.3 Logic

Logic makes up the basic aspects of an expert system. Logic is a broad term that encapsulates various methods and structures, whose aim is to represent human thoughts and reasoning[3]. It serves as the basis for inference and arguments, and there are many different forms of logic systems, such as:

- ***Informal Logic*** - is an *informal* logic system concerning the study of natural language arguments and is very subjective.
- ***Propositional Logic*** – is a *formal* logic system about the study of the conjunction of various axioms, propositions, statements or sentences in order to derive more complicated conclusions, as well as define logical relationships and properties.
- ***Predicate Calculus*** – extends on propositional logic and deals also with predicates and quantifications. A *predicate* is a statement, which works upon an element (or a group of elements) and is used to group elements that contain certain properties. A predicate is

[3] The word *logic* in itself is derived from the Greek *logike,* which means "possessed of reason, intellectual, dialectical, argumentative" and *logos¸* which is the Greek word for "thought, idea, argument, reason", among other representations. (http://en.wikipedia.org/wiki/Logic)

similar to a function in a computer program that returns a true/false value, and is also called a **propositional function**. *Quantifiers* are operators that are used to represent only a certain quantity of items from a set.

- **Boolean Algebra** – Invented by George Boole (Boole, 1848); expands on propositional logic and provides an algebriac-like system for dealing with logiacal statements. Used extensively in electronic computers.

1.4 Ontologies

An ontology is a formal specification of all the objects and properties related to a domain, and relationships between them.

Components of an ontology:

- **Individuals** are the most basic element in an ontology.
- **Classes** encapsulate a related set of objects, and define what an object is made up of.
- **Attributes** define characteristics about objects (or classes).
- **Relations** are ways in which objects are related to one another.

Ontologies are needed because it allows systems to know exactly what they are reasoning about, and avoid ambiguity. Since the aim of *Paramenides* is to cater for any type of knowledge base, each knowledge base must act upon an ontology in order to define its 'world'.

1.5 Knowledge Management

Knowledge is an extremely important asset. However, having enormous amounts of unstructured knowledge can pose several problems. Sometimes, work is repeated simply because it is impossible to make 'sense' out of the available knowledge, even though such knowledge exists and could have been used in the first place to reduce costs and time. Knowledge Management involves giving structure to information and thus make it usable. This process can be split into four cycles for expert systems:

Knowledge Acquisition

Its main aim is to retrieve information from different data sources, and convert it into a form that is more useable by the system.

This is normally done directly via a user interface used directly by the expert, or by a *knowledge engineer*, who converts the knowledge of the expert into a more suitable format for the system.

Expert systems rely greatly on **tacit knowledge** [4] **.** Tacit knowledge is that knowledge which is available only to an individual and only that individual holds that particular knowledge. Knowledge Acquisition aims at exposing this type of tacit knowledge and converts it into **explicit knowledge.**

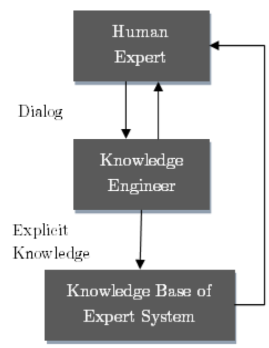

Figure 1-2: Knowledge Acquisition process (Giarratano, 2005. p. 10)

Knowledge Representation

This process involves converting knowledge into a formal representation, and can be done by various methods:

- **Rules:** a formal representation of one logical statement and creates a relationship between facts in the knowledge base. (Cawsey, 1998)
- **Semantic network:** is a labelled, directed graph that represents information

[4] Tacit knowledge can be considered knowledge that people are not aware of, or knowledge that cannot be easily explained (transferred) to other people.

about facts and the relationships between facts. The nodes represent the *objects* and *facts* of a knowledge base, while the edges represent relationships between them. (Sowa, 1987)

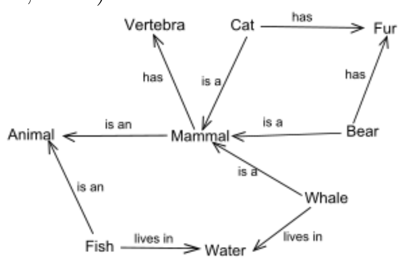

Figure 1-3: Visual depiction of a semantic net5

- **Associative Network:** similar to a semantic net, but nodes can be linked with other nodes to share information between them. For example, if we know that an animal can move, and we also know that a human is an animal, we can easily deduce that a human can also move. Today, this is now known more conventionally as *inheritance.*
- **OAV[6] Triples:** a general purpose data-structure that stores facts as a relationship between an object such as *cat*, an attribute such as *colour* and a value such as *white.*
- **Frames:** define a *skeleton* to describe an object. These are used to represent a narrow subject in detail, and are especially useful for describing real-world objects that contain various properties, like mechanical devices. A frame is made up of both *member links* and *subclass links.* Member links can be thought of representing a '*has-a'* relationship, while subclass links represent an '*is-a'* relationship.

Knowledge Inference

There are mainly two types of inference in an expert system – *Backward chaining* and *Forward chaining.* Backward chaining is a method of reasoning that starts with a goal (hypothesis) and works from it backwards until a solution is found and is especially useful for troubleshooting problems.

Forward Chaining is a form of *greedy* method of reasoning which considers any current known facts and tries to obtain as much information as possible by firing rules. This method is useful when one wants to diagnose a problem:- One does not know the problem, but knows only of the symptoms that he is experiencing. (Forsyth, 1986)

Uncertainty

Most real-world problems do not consist of discrete answers (yes/no). Sources of uncertainty in problem solving mainly are due to imperfect knowledge about the domain. (Jackson, 1999)

- **Bayesian Probability:** extends on probability theory and uses information about events based on certain experience. It uses probability values for each event outcome, and tries to maximize the returns. (Howson, 2005)
- **Dempster-Shafer Theory:** extends on Bayesian probability, but also allows degrees of beliefs in a statement, as opposed to more formal probability values for the outcomes. (Shafer, 2002)
- **Certainty Factors:** are a method for combining both belief and disbelief into a single value. Certainty Factors for different facts can be combined together to result in one CF, and hypothesis can be ranked according to their likelihood.

2 Aims and Objectives

The main aim of this project is to design and implement an expert system that:

- Is accessible easily to a large user-base through the use of the Internet.

[5] Reproduced from:
http://en.wikipedia.org/wiki/Semantic_net
[6] Object-Attribute-Value

- Has an easy-to-use user interface that does not require users to be particularly computer-savvy to use it.
- Is not limited to one particular domain, but serves as a generic expert system that can include various distinct knowledge bases to cater for large amount of domains.
- Able to perform both backward and forward chaining.
- Can easily explain its reasoning by a graphical representation to make it easier for users to understand the conclusions.

3 Design and Implementation

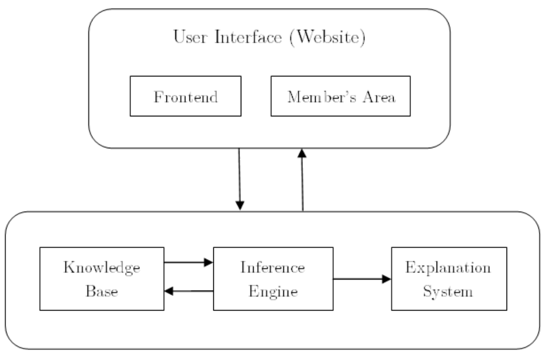

Figure 3-1: Block diagram of an expert system

The **knowledge base** contains the rules, and is used by the inference engine to retrieve facts based on current conditions. One can store the KB as *flat files, xml files* or in a *DBMS*. The DBMS is the system of choice because it abstracts the details of storage, is very efficient and easily searchable.

During research, we have come upon various existing **inference engines.** We have analysed in-depth such systems, in order to aid us in the design stage like CLIPS, Jess and NxBRE. We will be implementing our own inference engine since we no existing system was suited for all our requirements.

Since the aim of *Paramenides* is to be a global resource, it's **user interface** will consist of a website in ASP.Net and AJAX. We have split the functionality in various stages, and the user interface will be kept as simple as possible in order to cater for a large number of users. ScriptSharp[7] will be used to aid in JavaScript development, while we will use Dojo Toolkit [8] for added JavaScript functionality and cross-browser compatibility.

The **Explanation System** is very important to explain the reasoning of the system to the user. The system will keep track of rules fired in a chronological order, as well as the *certainty factors* of the facts in order to create a visual-representation of its reasoning. This will be presented to the user through a component created in *Adobe Flash*[9], and will allow the user to easily follow up on the reasoning process.

Evaluation and Results

Evaluation was based on the three main aspects of *Paramenides*: Backward Chaining, Forward Chaining, and Rule Management. We have

[7] http://projects.nikhilk.net/ScriptSharp

[8] http://www.dojotoolkit.org/

[9] http://www.adobe.com/products/flashplayer/

References

Boole, G. (1848). The Calculus of Logic. *Cambridge and Dublin Mathematical Journal , 3.*

Cawsey, A. (1998). *Rule-Based Systems.* Heriot-Watt University, Department of Computing and Electrical Engineering, Edinburgh.

Colin Howson, P. U. (2005). *Scientific Reasoning.* Open Court.

Jackson, P. (1998). *Introduction to Expert Systems.* Addison-Wesley.

Joseph C. Giarratano, G. D. (2005). *Expert Systems: Principles and Programming.* Course Technology.

Richard Forsyth, C. N. (1986). *The hitch-hiker's guide to artificial intelligence.* Taylor & Francis.

Shafer, G. (2002). *Dempster-Shafer Theory.*

Sowa, J. F. (1987). *Semantic Networks.*