South African Journal
of Information Management
[DoE accredited]

Peer Reviewed Article

Vol.3(2) September 2001

Investigation of a Web-based expert system shell

Dieter Vogts
Vista University (Port Elizabeth)
VOGTS-D@pelican.vista.ac.za

Contents

1. Introduction

The identification of objects has always been of importance to people, whether it be the identification of foodstuffs, poisonous plants, illnesses or the reason that a computer is malfunctioning. People are able to make these identifications based on knowledge gained from education and experience. Amazingly enough, people are able to make identifications even though no two objects are exactly alike (inexact knowledge) and even if they do not have all the information about an object available (incomplete knowledge).

The knowledge required to make certain identifications is often very specific, comprising an in-depth knowledge of the domain in which the identification is to be performed. The acquisition of this knowledge may take years and be very expensive. Through the use of expert systems, it is sometimes possible to computerize this knowledge, allowing the computer to make identifications at approximately the same level as an expert. Non-expert users are then able to conduct consultation sessions with the expert system in order to make use of the expert system's knowledge.

The medium of the WWW provides many opportunities for expert systems. Quick access can be made to world-wide distributed data and information, allowing for resources to be used from various locations and allowing anyone with an Internet connection to use the expert system. Multi-media based communication permits the use of many different styles of human-computer interaction. The expert system's knowledge bases are always up to date, eliminating the need for sending out updates and patches. All these, and the fact that the use of CGI applications make the expert system hardware independent, make that the WWW provides an exceptionally versatile and useful medium.

top

2. Identification problem

In the identification problem (also known as the classification problem), the domain of knowledge in which the classification is to be performed has a finite number of classes (or types) to which each object in the domain belongs. A finite set of properties is assumed to exist that can be used to differentiate between the classes. The exact number of properties that exist varies, but typically there are in the tens to hundreds. Properties have values with which they can be assigned. A set of property-value pairs describes an object. An object need not have a property-value assignment for each and every property. This is typically the case for objects on which certain features are not present at all.

For example, the domain of knowledge chosen for the test case of the expert system was that of the identification of a number of South African *Encephalartos* (cycad) species. Each specie that could be identified was a class and a particular cycad was an object. Of the 80 properties used in an affinity study mentioned in Osborne (Osborne 1990), only 19 were used. These included the curve direction of the leaves, leaf colour, cone text and seed colour. Values that could be used in an assignment were enumerated values, such as light green, blue green and dark green for leaf colour. A particular cycad could be described by a set of property-value assignments, such as {leaf length = 50-100 cm; leaf width = 10-20cm; leaf colour = dark green; rachis colour = light green}.

Given a sample object that belongs to one of the classes, the identification problem is to establish the assignments for the properties of the object in order to determine the class or most likely classes to which it belongs. For example, given a sample cycad, determine what to ask the person about it to be able to identify it (as quickly and accurately as possible). For some domains of knowledge, it is important to try and ask for as few assignments as possible, as the determination of an assignment can be expensive. For example, if the domain of knowledge were the diagnosis of an illness, the properties could include blood pressure, cholesterol number and blood cell count. The determination of these values can be expensive and time consuming, therefore the fewer assignments asked for, the better (assuming the identification is correct).

The problem of identifying an object is complicated by a number of factors:

- Incomplete information
- Inexact information
- Not all the objects in the domain may be known (even though they are finite).

Incomplete knowledge has to do with the problem that not all the necessary knowledge may be known about an object in order to make the classification. For example, given only a leaf of a cycad, it is still possible to draw some conclusions about the class of the cycad.

Inexact knowledge has to do with the fact that no two objects are alike and that, when querying someone for information about an object, they may give answers that are inaccurate. For example, a person holding two different cricket balls could classify them as being cricket balls, even though the balls might have slightly different masses and be different shades of red. Also, what may seem like a light green leaf to one person may be seen as a grey green leaf by another. Therefore there is a degree of fuzziness that must be catered for.

And lastly, not every single object in the domain may be known. More specifically, not every set of property-value assignments is known for objects of every class. To overcome this, generalizations need to be made about property-value sets of objects in a particular class.

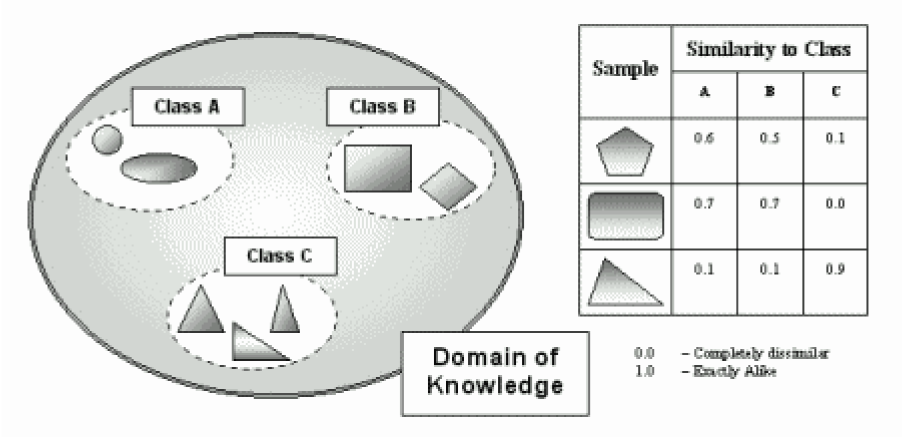
There are a number of methods that have been used in an attempt to solve the identification problem. Some approaches include using SQL to query relational databases (XID, 2000), rule-based expert systems (Lucas and Van der Gaag 1991), fuzzy associative memories (Kosko 1992), statistical pattern recognition (De V. de Kock 1991), predicate logic (Durkin 1996), proximity-based classifications (López-Aligué *et al*. 1991) fuzzy logic (Kosko 1992) and neural networks (Michie 1994) to name a few. Each uses a slightly different approach when performing identification, but all work on the assumption that the type an object belongs too is the one that is 'closest' to it. The exact definition of 'closest is determined by the method used. Each method is suitable under certain conditions.

The expert system shell developed uses a technique based on the observation that the values of properties are all similar to a certain degree and classes can be described by a representative number of samples from them. Using these heuristics as building blocks and some concepts from some of the aforementioned methods, a technique was developed and implemented called the similarity classification method (Vogts 1998). Using the similarities of the various properties of objects, it is possible to compare two (or more) objects and determine how similar they are. The knowledge of the domain is acquired by providing samples of various types and these are used to determine to what degree another object could be classified as the class.

For example, given a number of *E. trispinosus* samples, and then to determine whether another cycad belonged to the *E. trispinosus* species or not, it would be compared with the samples known to belong to the species. The results of the comparisons are combined to acquire an overall degree of similarity to the class. A value of 1.0 indicates total similarity, while a value of 0.0 indicates complete dissimilarity. Values in between 0.0 and 1.0 indicate partial similarity. It is possible for a given sample to be similar to various classes to differing degrees. There is normally, however, one class that is a lot more similar to the samples than others. The similarity classification method attempts to either prove or disprove the similarity until some termination conditions are achieved.

Figure 1 shows a very small domain of knowledge of simple shapes grouped into classes A, B and C. Three samples are examined and their degrees of similarity with each class displayed. It is interesting to note that even though certain objects, such as the pentagon and rounded rectangle, have not explicitly been said to belong to a particular class, the similarity classification technique can still make some observations about them.

Figure 1 Example of a domain of knowledge of simple shapes



Sample	Similarity to Class		
	A	B	C
⬠	0.6	0.5	0.1
▢	0.7	0.7	0.0
△	0.1	0.1	0.9

0.0 – Completely dissimilar
1.0 – Exactly Alike

A consultation session normally follows the following steps:

1. Start off with a set of initial assignments (typically none).
2. Determine the property that has not already been assigned that will on average prove as few classes similar to the object and disprove as many classes similar to the object.
3. Query the user for the assignment.
4. If the termination conditions have not been met, go to step 2.
5. Display the classification and any other related information.
6. Terminate the consultation session.

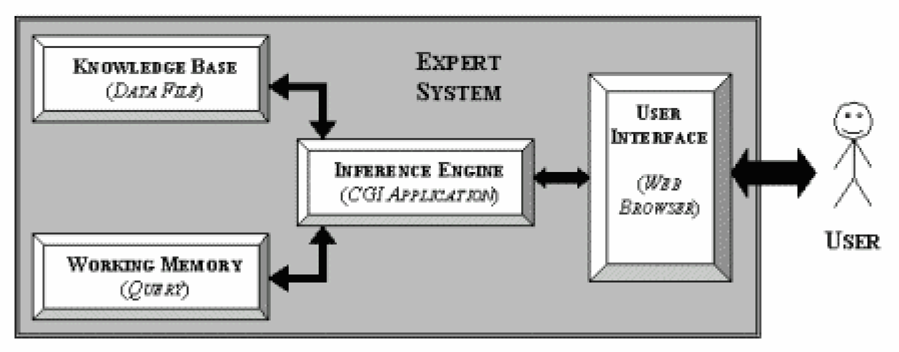
At any stage, it is possible to go back and change an assignment made so far.

top

3. Web expert system shell design

A typical expert system consists of a number of interacting parts. Figure 2 shows a simple block diagram of the constituting parts.

Figure 2 Expert system block diagram and the corresponding physical counterparts (in brackets)



The knowledge base is the repository of knowledge that encapsulates the knowledge required by the system to exhibit the problem-solving ability of the system. It contains all the heuristics, hard facts and heuristic knowledge that an expert knows about the domain. In the implementation of this expert system, the data file containing the information for a knowledge base is the physical manifestation of the knowledge base.

The working memory contains all the information and conclusions that have been derived during the consultation session, as well as the information that was retrieved from the user of the system during the process. The physical counterpart in this expert system shell is the query that is submitted by the Web browser.

The inference engine, on the other hand, is the part of the expert system that utilises the knowledge in the knowledge base and the working memory and is able to infer new information or make decisions based on the information that it knows. The physical manifestation of this is the CGI application that is executed by the Web server when a query is obtained.

Fr the system to interact with the expert system, a fourth module is needed, namely the user interface. This is the window through which the system is able to return information to the user and query him/her for further information. It is also the part of the system that the knowledge engineer will most likely want to change depending on the prospective users. Therefore a very flexible way of specifying the user interface is needed. HTML and Internet browsers provide an exceptionally flexible, elegant and easy-to-use method for this. The inclusion of multimedia such as images, sound and video are trivial to effect, provided that the browser has the correct plug-ins available.

A consultation engine (the name given to a knowledge base and inference engine) is consulted by passing a query to the CGI application. A complete representation of the working memory of the consultation needs to be passed in as information about requests to the server are not stored by the application. A typical request to the consultation engine looks something like this:

http://a.site.somewhere/cgi-bin/zce.dll?:kb=cycads&leaf+colour=light+green&:th=text+only

There are a number of different parameter types that can be specified in the query. Parameters are separated from one another by an ampersand, '&'. Those starting with a colon, ':', are system parameters and are used to set various aspects of the consultation. All queries must include a *:kb* parameter that specifies the name of the knowledge base to be used in the consultation. An optional system parameter is *:th* that specifies the theme (see below) to be used when generating the output page. If a theme is not specified, then the default theme of the knowledge base is used instead. Property-value pairs are specified as <property name>=<value> pairs with spaces replaced by pluses. If the same parameter is specified more than once, the last occurrence of it is used. This allows for the easy replacement or deleting of parameters. The following queries are equivalent:

leaf+colour=light+green&leaf+length=10+-+20&leaf+colour=dark+green
leaf+length=10+-+20&leaf+colour=dark+green
leaf+colour=dark+green&leaf+length=10+-+20

top

4. Themes, templates and resources

Before a consultation is begun, the interaction style or theme needs to be specified. A theme is a collection of templates that specify how a user and the expert system will interact. This is particularly useful for customizing the expert system to different user groups and needs. Common reasons for having different themes include:

- Interacting with the users in their own language
- Providing special interfaces for people with visual and auditory disabilities
- Scaling the interface up or down for performance reasons

Templates are used to describe a particular part of the entire user interface and the method of interaction with the user of the consultation engine. Every class and property in the expert system has a template associated with it for each theme. The template is simply a text file describing the document layout in HTML. Special tags are included that the system fills in with data in response to a query.

The special tags used begin with a hash symbol that distinguishes them from normal HTML tags. When a Web browser encounters these, it normally ignores them, but when the consultation engine processes them, it typically substitutes them with other HTML or performs an operation.

There are a number of different types of tags:

- General information
- Focus selection

- Navigational
- Field information
- Resources
- Iteration

Each type has a different purpose and eventual result. General information tags allow the client to request basic information about the consultation engine such as the name of the knowledge base, the date of creation, name of the author and version number. Focus tags allow the current data list being examined to change to another. Navigational tags allow for the navigation through the current data list. Field information tags allow the client to request different types of data (typically calculated) of the current item being examined. Resource tags contain information about various resources associated with a particular item in the current data list. Iteration tags repeat whatever is contained between their <#begin> and <#end> tags a number of times (typically once for each item in the data list).

For example, the template below

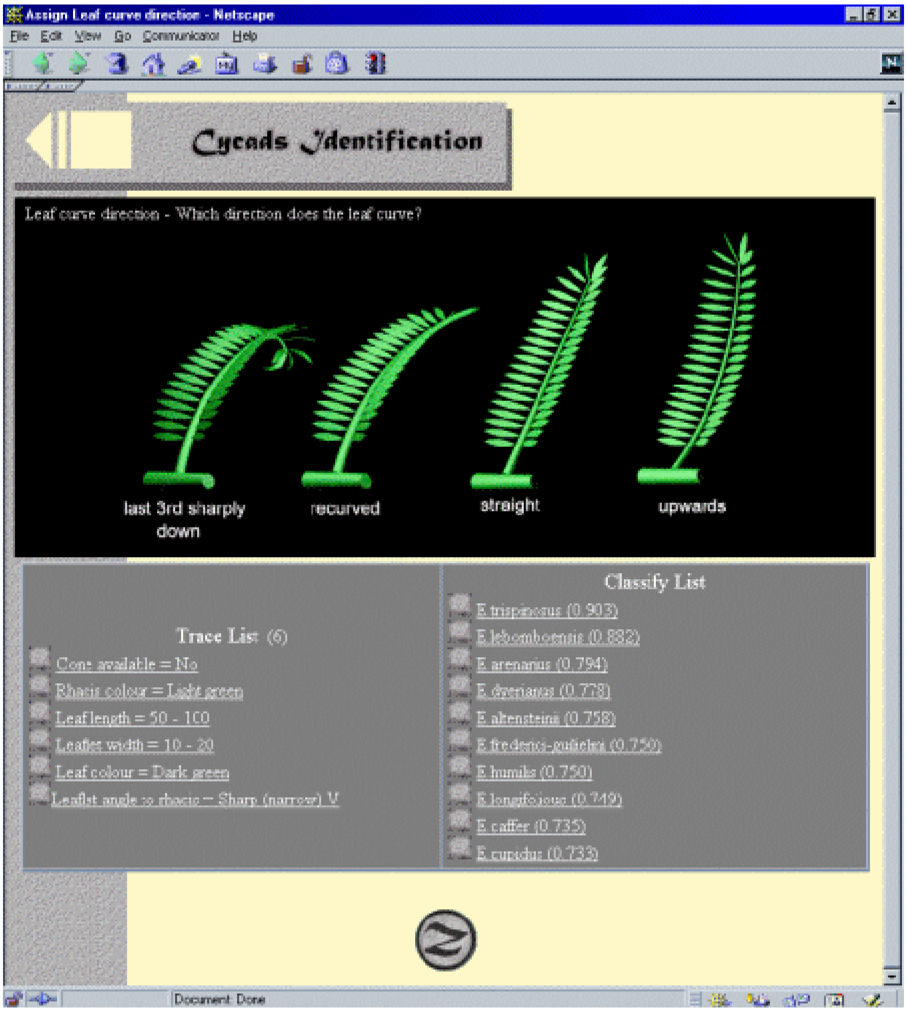
```
<html>
<title>
<#kb> consultation session v<#version>
</title>
<body>
<b>Question</b><br>
<#assign>
<#property resource=English_Question><br>
Select your answer from one of the following:<br>
<#first>
<#begin>
<li><a href=<#value field=url>><#value resource=English_Text></a>
<#end>
</body>
</html>
```

once processed, could be transformed into the HTML document below.

```
<html>
<title>
Cycads consultation session v0.1
</title>
<body>
<b>Question</b><br>
What is the shape of the sample plant's leaf?<br>
Select your answer from one of the following:<br>
<li><a href=http://www.asite.za/cgi-bin/zce.dll?:kb=cycads&Leaf+shape=Rounded>
Rounded</a>
<li><a href=http://www.asite.za/cgi-bin/zce.dll?:kb=cycads&Leaf+shape=Pointed>
Pointed</a>
</body>
</html>
```

Figure 3 shows a screenshot of an actual template that has been filled in with data by the consultation engine.

Figure 3 Example of a filled-in template



Assign Leaf curve direction - Netscape

Cycads Identification

Leaf curve direction - Which direction does the leaf curve?

Trace List (6)

- Cone available = No
- Rhachis colour = Light green
- Leaf length = 50 - 100
- Leaflet width = 10 - 20
- Leaf colour = Dark green
- Leaflet angle to rhachis = Sharp (narrow) V

Classify List

- E.trispinosus (0.903)
- E.lebomboensis (0.882)
- E.arenarius (0.794)
- E.dyerianus (0.778)
- E.altensteinii (0.758)
- E.friderici-guilielmi (0.750)
- E.humilis (0.750)
- E.longifolius (0.749)
- E.caffer (0.735)
- E.cupidus (0.733)

Document: Done

For the templates to fill in useful information, the classes, objects, properties and values in the expert system have named resources that are substituted into the document when resource tags are encountered. A resource is simply a text string that is user definable.

Uses of resources include:

- providing information in a particular language
- a media URL
- information about image maps
- JavaScript code (or parts thereof) to be placed in the final document
- a VRML representation of a class, object, property or value
- links to other sites containing more information.

To create a new theme, all that is required is to create the new templates and add them to the knowledge base. If the templates use resources that do not exist in the knowledge base, then these must be added as well.

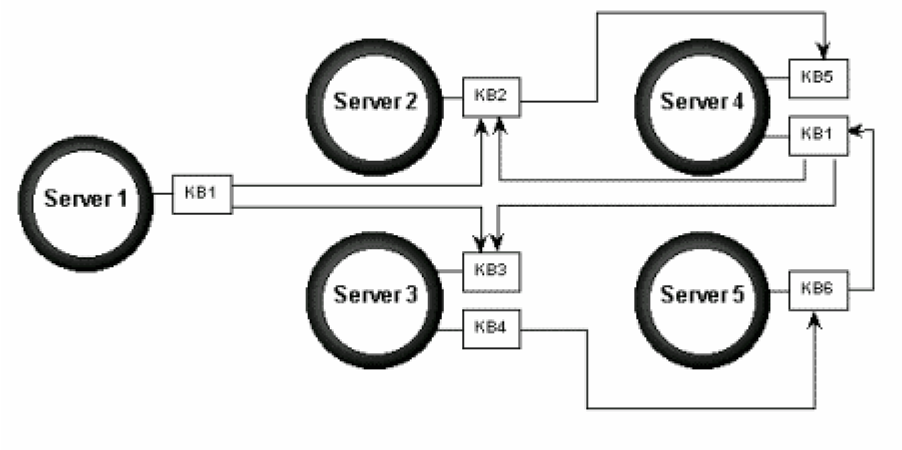
During a consultation session, it is easy to change to a different theme. This is achieved by appending the system parameter :*th*=<new theme name>.

top

5. Consultation engine networks

Owing to the design of the expert system shell and the WWW it is possible to connect various consultation engines together. Whether the consultation engines reside on different servers or the same server is inconsequential (see Figure 4).

Figure 4 Example of a consultation engine network



The classification (or partial classification) of one consultation engine may be used as input into another consultation engine. This can be done, as a consultation engine will only examine and operate on parts of the query that are pertinent to it. All other information is left untouched. Thus it is possible to send a request for another consultation engine, passing all the information gathered about the current consultation, but appending additional information to get the new consultation running.

For example, a 'general practitioner' consultation engine could determine that a patient has a very good chance of having a heart disease. This consultation engine could then invoke a 'heart specialist' consultation engine to determine if the patient is indeed suffering from a heart disease or not and, if so, what it is.

The only proviso is that various knowledge bases should not share the same name space for properties as this may cause some confusion. Overcoming this is fairly straightforward if a naming convention of *knowledge_base_name.property_name* and *knowledge_base_name.class_name* is used to identify properties and classes respectively.

An ability of being able to create networks of consultation engines is that larger knowledge bases can be built up of smaller logical knowledge bases. There are many advantages to doing this, the most obvious being cost, speed of execution and manageability.

top

6. Into the future…

There are many practical applications for an expert system of this nature. The fact that it can be accessed remotely means that community computer centres of underprivileged areas could become a place of more than just education. Interest has been expressed in the system by a South African telecommunications company for the use of the expert system in a hand-held computer to be used by technicians who need to repair lines, relays and switches. Further research is being conducted into other possible uses and improvements of the current system.

top

7. Acknowledgements

This paper is based on an MSc dissertation entitled Design of a Multimedia Environment/Expert System Shell for the Identification Problem, at the University of Port Elizabeth (supervisor: Professor G. de V. de Kock, Department of Computer Science).

top

8. References

1. De V. de Kock, G. 1991. Expert system shell for the classification problem. *South African Journal of Science*, 87:179-182.
2. Durkin, J. 1996. *Expert systems design and development.* New Jersey: Prentice Hall.
3. Kosko, B. 1992. *Neural networks and fuzzy systems: a dynamical systems approach to machine intelligence.* New Jersey: Prentice Hall.
4. López-Aligué, F.J., Acevedo-Sotoca, I. and Jarmillo-Moran, M.A. 1991. The fuzziness of fuzzy partitions. *Pattern Recognition Letters,* 12:265-271.
5. Lucas, P. and Van der Gaag, L. 1991. *Principles of expert systems*. New York: Addison-Wesley Publishing.
6. Michie, C.J. 1994. *Machine learning, neural networks and statistical classification*. London.
7. Osborne, R. 1990. Encephalartosumbelunziensis. *Encephalartos*, 21:3-9.
8. Vogts, D. 1998. Design of a multimedia environment/expert system shell for the identification problem. Unpublished MSc dissertation. Port Elizabeth: University of Port Elizabeth.
9. XID Services Inc. 2000. Expert identifications systems. Available at WWW: http://www.xidservices.com/.

Disclaimer

Articles published in SAJIM are the opinions of the authors and do not necessarily reflect the opinion of the Editor, Board, Publisher, Webmaster or the Rand Afrikaans University. The user hereby waives any claim he/she/they may have or acquire against the publisher, its suppliers, licensees and sub licensees and indemnifies all said persons from any claims, lawsuits, proceedings, costs, special, incidental, consequential or indirect damages, including damages for loss of profits, loss of business or downtime arising out of or relating to the user's use of the Website.

top

ISSN 1560-683X

Published by InterWord Communications for the Centre for Research in Web-based Applications,
Rand Afrikaans University