w illiams

6èmes Journées Internationales

LES SYSTEMES EXPERTS & LEURS APPLICATIONS

Palais des Papes - Avignon, France
28-29-30 avril 1986

6th International Workshop

EXPERT SYSTEMS & THEIR APPLICATIONS

Palace of the Popes - Avignon, France
April 28, 29 & 30, 1986

VOLUME I

Agence de l'Informatique
Établissement Public National

SESSION 2A

Acquisition, Représentation & Validation des Connaissances – 1

Knowledge Acquisition, Representation & Validation – 1

GEM: A MICRO-COMPUTER BASED EXPERT SYSTEM FOR GEOGRAPHIC DOMAINS

Graham J. Williams
Department of Computer Science
Australian National University
G.P.O. Box 4, Canberra, Australia 2601
Phone: + 61 62 494043 CSNet: gjw%anucsd.oz@csnet-relay

Paul M. Nanninga and J. Richard Davis
Division of Water and Land Resources
Commonwealth Scientific and Industrial Research Organisation
G.P.O. Box 1666, Canberra, Australia 2601

Abstract

GEM is a rule-based expert system shell designed for application to geographic-type problems. It was developed as a micro-computer based system allowing for ready access to its knowledge-base by both experts and users, initially in the domain of fire prediction in a National Park. In the context of geographic problem solving, the application of rules can be restricted to certain geographic regions and parameter equivalencing is employed to allow spatially invariant information to be represented and used. In the context of problem solving in general, the idea of multiple sources of information is used to allow a variety of methods to be employed in the problem solving process. A number of improvements are discussed including the use of frames to represent declarative knowledge and equations to represent arithmetic knowledge, and the need for computer assistance with knowledge acquisition.

Keywords: Rule-based systems, geographic problem solving, classification problem, PROLOG, micro-computer, fire management.

This research was supported by The Australian National Parks and Wildlife Service, and carried out at The Division of Water and Land Resources, Commonwealth Scientific and Industrial Research Organisation, Canberra.

1. Introduction

This paper describes an expert system shell, called GEM, which has been built to record and apply knowledge for geographic domains. Expert systems are characterised by the separation of the inference engine (the means of applying knowledge to a problem in the domain) from the store of knowledge (the knowledge base), and by the use of heuristic knowledge, commonly in the form of production rules (Davis and King, 1984). These rules are typically represented in the form "IF X is true THEN conclude Y to be true"; X may contain a number of conditions. This results in the knowledge being encoded as many independent chunks or *parcels* of knowledge. Such knowledge is used by the inference engine to purposively determine new information until some item of information required by the user (the goal) is reached. General introductions to the topic of expert systems are now widely available (Hayes-Roth et. al., 1983, Waterman, 1986) and we will not describe them further here.

The knowledge used in solving problems in geographic domains has a number of distinguishing characteristics. In particular, spatial information commonly needs to be used, and the system is often required to make use of knowledge from a number of local geographic regions, when solving a problem in some specific region. Expert systems in the geographic domain are rare; examples are a flood warning system (Cuena, 1983), an environmental assessment system (Pereira et. al., 1982), a hybrid expert system for predicting faunal population dynamics (Starfield et. al., 1985), and various attempts to combine expert system technology with geographic databases (Smith, 1984). In addition McDermott and Davis (1984), have investigated a means of representing geographic-type information. However, few of these programs possess features specifically designed for geographic applications. GEM represents a preliminary attempt to develop an expert system shell (i.e. general purpose inference engines able to access knowledge bases constructed for specific problems) for such applications.

2. Fire Management

One geographic problem which GEM has tackled is the prediction of the effects of fires in an Australian National Park. Although in this paper we concentrate on GEM, examples are drawn from its application to this domain; the application being more fully described in (Davis et. al., 1986).

The management of wildfires in Australia is a typical problem requiring geographic information. Such fires often spread over vast areas, and the rate at which this spread occurs depends upon many factors, including the distribution of vegetation, the slope and aspect of the land encountered, and changing meteorological conditions across the area (Walker et. al., 1985). Other physical fire parameters such as the height of the flames are also affected by geographically distributed data.

Kakadu National Park in the Northern Territory, Australia, is one of a number of wilderness areas listed on the World Heritage List. The park is owned by the indigenous aboriginal population, but managed under lease by the Australian National Parks and Wildlife Service (Australian National Parks and Wildlife Service, 1980). Large areas of the Park are burnt each year by wildfires. Such fires are started by both natural means and by man, and the periodical controlled burning of fuel loads is one of few management options open to the Park rangers. Controlled burns have also traditionally been used by the aboriginal population to manage the environment to suit their needs. With European settlement, though, these traditional practices have been abandoned and, as a consequence, both the flora and the fauna of the area have changed.

One of the considerations of the rangers in planning their controlled burns is to establish the effects of these fires on the biota of the park. The actual timing of these burns is crucial, as fires lit too early in the dry season will not propagate, and burning too late will lead to major fires which will burn for many weeks, possibly further altering vegetation structure and affecting faunal communities.

With experience, a ranger can predict suitable times for control burning. The prediction is largely based upon observation of the fuel load, the prevailing and immediately past weather conditions, and the past fire history of any particular region in the park (Hoare, 1985). In effect over several years the ranger has learnt a series of rules of thumb about the effects of fire under diverse conditions.

But park rangers are not forever. A ranger's knowledge is built up over many years of experience in a particular region, and a new ranger needs several years to become competent in fire management. There are no process models, verified for the Park, to tell the ranger when to light fires; i.e. there is no deep understanding, at present, of the processes involved in such systems. Consequently, expert systems offer one of the best approaches to codifying the accumulated expertise of the Park rangers for educational purposes. They also open up another possibility, and that is the transference of knowledge about fire behaviour and effects from academic experts to Park staff, experienced and novice. At present the GEM shell is being applied to this latter task in Kakadu National Park.

For this application of GEM, the land systems and land units of Story et. al. (1969) and Story et. al. (1976) are used as the geographic regions. Some typical parameters used in the application are the season, the wind strength, and the humidity. The determination of values for parameters such as the fire danger rating and the flame height, are typical goals.

3. Architecture of GEM

3.1. Overview.

A rule-based system was chosen as the basis for GEM for a number of reasons. Firstly, it provides a natural formalism in which domain experts (who are often computer illiterate) can express much of their knowledge (Newell and Simon, 1972). This implied that GEM needed to incorporate a high-level editor with which an expert could interact. We will not discuss the editor module of GEM in this paper, but will concentrate on the structure of the knowledge base and the operation of the inference engine. Secondly, by formalising their knowledge in this way the domain experts are able to pinpoint the areas in which their knowledge is deficient, and can compare their knowledge with that of others. And thirdly, expert systems provide a conceptually simple method for experimenting with knowledge.

Several requirements needed to be taken into account in the design of the GEM system. Firstly, the nature of the geographic domain required that both local and global knowledge be represented. This entailed the use of spatially invariant parameters, so that, rather than a parameter having a single value, it could have a number of values, each corresponding to a different region or regions of the Park. The individual rules also need to be associated with the regions in which they are applicable. Secondly, because the first application of the shell in Kakadu National Park required that the system be used on a micro-computer giving real time response, it must be portable and efficient. Thirdly, the problem of estimating the damage caused by fires could be formulated as a "classification" problem (Weiss and Kulikowski, 1984); a problem type well-handled by rule-based systems. Consequently a standard classification shell program, EMYCIN (Mackenzie, 1984), was modified to incorporate the spatial features (Davis and Nanninga, 1985). But this was ultimately fc d o violate the second requirement; being too large and slow to be practically usef. Subsequently a portable geographic shell, GEM (GEOMYCIN for Micro-Computers), was developed, using PROLOG (Clocksin and Mellish, 1981). Its architecture is summarised in Figure 1.

GEM consists of several conceptual layers, with the *Domain Files* forming the base layer, and representing the knowledge base. Access of the domain files is through the *Domain Interface Modules*, which are called upon by the *Consultation Module*. The overall operation of the system is directed by the *Executive*. An editor is also provided to allow any part of the knowledge base to be easily modified by the expert.

The domain files contain the domain dependent information of the system, and consist of five files. A sixth interface and domain file, the algorithmic base, is currently being implemented. The rule base is a store of knowledge in the form of production rules; the geographic database is a store of relatively static information (i.e. invariant between consultations) about each region; and the so called user base is a pseudo-file which is used here to represent the user of the system. The information stored in this pseudo-file consists of parameter values which the user supplies to the system. The parameter base contains information describing the individual parameters known to the system in this domain and the region base lists the various regions constituting the management area. Such a disaggregated approach to the storage of information facilitates the representation of a wide range of knowledge since the expert is not restricted to a single representation scheme.

The parameters and rules are described in the next two sections of the paper. An example of a single record, in the Kakadu database, each record consisting of three data items for each region, is:

Kay1

Fuel Weight:	Moderate
Fuel Type:	Annual
Description:	Tall open forest

The format file is not a domain file in the sense of being a store of information about the domain, but rather, specifies the structure of the domain files. This file is used to guide the access to the domain files by the domain interface modules.

The domain interface modules encode the basic procedures which make use of the knowledge stored in the domain files, especially representing the various methods known to the system for determining values for parameters. The rule module, given a parameter and a region, will retrieve those rules in the rule base which can be used to determine a value for that parameter in that region. The database lookup module provides a front end to the database, and will retrieve the value of a specified parameter in a specified region, if such a value is stored in the database. The user interface is simply a collection of routines for asking the user questions.

Each attempt by the system to determine the value of a parameter in a specified region requested by the user (a goal parameter) is termed a consultation, in keeping with the MYCIN terminology (Shortliffe, 1976). The consultation module controls each consultation. During any consultation, values for other parameters may need to be determined, and the values of parameters in other regions may also need to be used. The consultation module will call upon a number of the domain interface modules, including the rule module, to obtain values for these parameters. The algorithm employed by the consultation module is

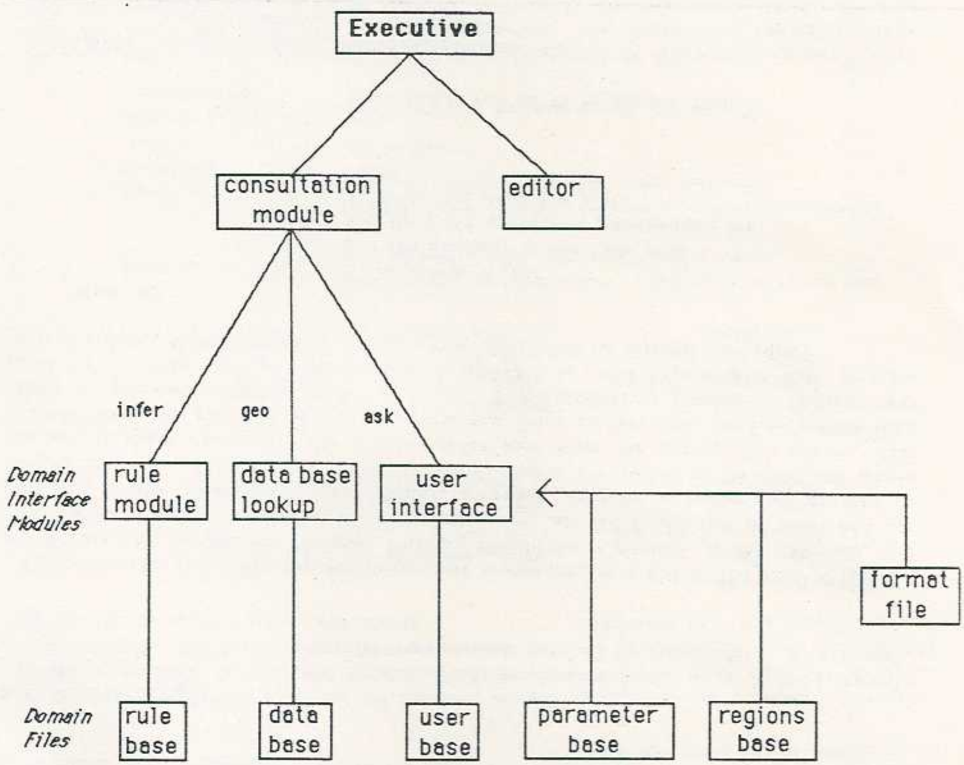

Figure 1: Architecture of GEM

detailed in section 3.4.

The executive directs the running of the system. This module provides options to allow the user to begin a consultation, to review previous consultations, to clear the memory of previous consultations, and to edit the domain files.

The following sections detail the principal components of the system.

3.2. Representing parameters.

During the process of determining a value for the goal parameter, the system may need to determine values for other parameters upon which this goal parameter depends. Most of the knowledge in the domain files is expressed in terms of parameters, all such parameters being defined in the parameter base.

Figure 2 illustrates the information that is recorded for two typical parameters, and also shows the structure used by GEM to record the parameter definitions as PROLOG clauses. The information is structured in a record with four components - Value Type, Equivalenced Regions, Source List, and Description.

Wind Strength

Value Type:	[calm, low, moderate, high]
Equivalenced Regions:	none
Source List:	[ask]
Description:	[0 kph, 0-6 kph, 7-16 kph, >16 kph]

Humidity

Value Type:	percentage
Equivalenced Regions:	[Kay1, Kay2, Queue1, Queue2]
Source List:	[infer, ask]
Description:	Relative humidity mid-afternoon

```
parameter('wind strength', [calm, low, moderate, high],
        none, [ask], ['0 kph','0-6 kph','7-16 kph','>16 kph']).

parameter(humidity, percentage, ['Kay1','Kay2','Queue1','Queue2'],
        [algorithm ask], 'Relative humidity mid-afternoon').
```

Figure 2: Representation of parameters used by GEM.

Value Type specifies which of three value types the parameter can have. These types are *positive integers*, *percentages*, and *enumerated types*. The first two types are self-explanatory; if an enumerated type is required, then the values that the parameter can take are listed in this component. Such a list defines an order on the values, which can be referred to with, for example, the "greater than" relation, as described below.

The Equivalenced Regions component of the record specifies the regions for which this parameter is known to take a common value. That is, if a value for the parameter has been determined in one region in the list defined in this component, then the consultation module will assume that this parameter will have the same value for all other regions in this list. The algorithm presented in Section 3.4 describes this procedure more formally.

The Source List is a central feature of the GEM system. It provides a mechanism for specifying both the domain files to be tried when the consultation module attempts to determine a value for the parameter, and the order in which they are to be tried. The codes *infer*, *ask*, and *geo* refer to the rule module, the user interface and the database lookup respectively. Whenever a value for a parameter is to be determined, the consultation module will access the first domain interface module listed in this field. If the domain file called by this interface module fails to determine a value for the parameter, then the next interface module on the Source List is accessed, and so on until all listed interface modules have been tried.

The Description component allows some English text describing either the parameter, or the meanings of the values if an enumerated type was specified in the Value Type component. Both uses of the Description component are illustrated in Figure 2. This component is used by the consultation module whenever the user requires further information about a parameter during a consultation.

3.3. Representing rules.

In the application to Kakadu National Park, production rules, as with many experts systems, provide the basic means for representing most of the knowledge. This though, need not be the case in other applications.

The production rules, like the parameters, are represented in GEM in the form of records, which have the components Number, Premise, Conclusion, Certainty, Author, Regions, and Justification (Figure 3). Three of these components are for descriptive purposes and are not used in the inferencing. The Number component contains a unique number used to identify the rule. The Author component simply names the author of the rule, and allows the user to check whose expertise is being called upon when the system offers advice. This is useful when, for example, the user suspects that some authors have more experience than others in particular aspects of the problem. The Justification component contains text justifying the rule, through, for example, a reference to some experimental data or some journal article. Figure 3 lists an example rule along with its internal structure.

Rule 32

Premise:	If the season is cool, and the humidity is less than 40%, and the wind strength is greater than low,
Conclusion:	Then the fire danger is high.
Certainty:	100%
Author:	Jamie Hoare
Regions:	[all]
Justification:	Experiments run on 15 September 1980

```
rule(32, [[season, is, cool], [humidity, less-than, 40],
        ['wind strength', greater-than, low]],
        ['fire danger', is, high, 100],
        'Jamie Hoare', [all],
        'Experiments run on 15 September 1980').
```

Figure 3: Representation of a rule.

The premise of the rule, that is the IF-part of the production rule, is stored in the Premise component. Each member of the list is a triplet (Mackenzie, 1984) which has the form [P,R,V], where P ranges over the known parameter names, R is one of the relations {less-than, less-than-or-equal, greater-than, greater-than-or-equal, is, is-not, is-one-of}, and V is either a value which conforms to the Value Type component of the parameter P or, in the case that R is the relation is-one-of, is a set of such values. The triplets are conjoined. For a premise to be evaluated as "true", it must be true that, for each triplet, P has the relation R to V.

The Conclusion component represents the THEN-part of the production rule. It consists of a single triplet, in which R must be the relation "is". Associated with this component is the Certainty component which is a percentage indicating the certainty with which the expert believes this rule to be valid in the regions specified by the Regions component. Together, the Conclusion and the Certainty component specify that if the Premise triplets are all true, then the conclusion parameter P can be asserted to take the value V with some certainty. The manner by which the consultation module handles certainty factors is as in MYCIN (Shortliffe, 1976).

The final component of the rule record is the Regions component, which describes those regions for which this rule is valid. This component can take the value "all" or contain a list of defined regions.

3.4. The consultation module.

Information about a number of parameters is obtained during a consultation with GEM. This information concerns particular regions, and might be values for the parameters, bounds on the possible values for the parameters, or values that the parameter can not take. Such information is recorded internally in two separate data structures. The first is represented as a PROLOG clause of the form:

possible-values(P,Rg,L,U,E).

Here P names a parameter, Rg names a region, L is a lower bound on the value of the parameter in this region, U is an upper bound, and E is a list of values which the parameter can not have in this region (i.e. exceptions). This structure forms the basis of an attempt to allow GEM to handle information that is inferred as a range rather than explicit values. At present, other features (such as conflict handling) need to be added to GEM before the full strength of the *possible-values* predicate can be exploited.

Such information is used by a number of modules of the system. The rule module uses it to determine if a rule has any chance of succeeding. Consider a rule whose premise consists of a number of triplets. The *possible-values* information for the parameter in each triplet is inspected in turn by the consultation module. If the range of any one of the *possible-values* will not permit that triplet to be evaluated as "true", then the rule is considered no further.

The *possible-values* information is updated whenever a domain interface module succeeds. As an example, if the rule illustrated in Figure 3 were to succeed during a consultation in the Kay1 region, then the *possible-values* clause for the fire danger parameter would become:

possible-values(fire danger, Kay1, high, high, []).

where it is assumed that no other information has been determined for this parameter.

The second important internal store of information in GEM is defined by the *determined* clause. This differs from the *possible-values* clause in that it is instantiated only when a unique value, rather than a range of values, has been determined for a parameter. The format is:

determined(P,Rg,V,C,S).

where P and Rg are as before, V is the value determined for P, C is the certainty associated with this value, and S is the source of this value. S may be a user's name (if the value was supplied by a user), a rule number, or the word *geo* (indicating the database). An example is:

determined(humidity, Kay1, 30, 80, Rule10).

which says that the humidity in Kay1 is 30%, with 80% certainty, and that this fact was derived using rule number 10. Such information is always checked whenever a value is required for any parameter for either explanation purposes or for inferring further facts. Because GEM is designed for micro-computers, when a parameter is equivalenced across a number of regions there will only be one *determined* clause recorded, implicitly representing a number of clauses, one for each region in that parameter's Region component.

A third point of discussion is the basic algorithm used by GEM to determine a value for a given parameter in a given region. Although some PROLOG terminology is used, its intention should be understandable to the general reader. The single, lower case letters, (e.g. p, r, etc) represent instantiated values, whereas upper case letters such as V, represent variables which are to be instantiated in the usual PROLOG way.

To determine the value of p in region r:

1. If *determined*(p,r,V,C,S) indicates that a value has already been determined for p in the region r, then go to step 6.

2. Otherwise, if a value for p has been determined in some other region, and this region is equivalenced to r for the parameter p, then go to step 6. This involves evaluating *determined*(p,R,V,C,S), and checking that r and R are equivalenced for p.

3. If step 2 fails, set α to 1 and use the α^{th} method specified on the Source List of p. This results in one of the following actions being taken:

infer	Search for rules in the rule base which can determine a value for p in the region r, and which are not ruled out by the current *possible-values* of the premise parameters. These rules are then evaluated in the order in which they are recorded. This may involve determining values for other parameters, entailing recursive evaluation of this algorithm. It is feasible that no rule may succeed.
geo	Lookup the database for a value.
ask	Ask the user for a value. The user has the option of replying with the answer *unknown*.

   If this step succeeds, go to step 6.

4. If the α^{th} method on the Source List fails and further methods are listed, then increment α by one and return to step 3.

5. Otherwise the system is unable to determine a value, and *determined*(p,r,unknown,_,S) is recorded. Exit.

6. If a new value has been determined, then record it in the *determined* predicate.

4. An Example Consultation.

Whilst it is important that the features of GEM be suited to classification problems in the geographic domain, it is equally important that the program's interactions with the user be clearly laid out, easy to quickly comprehend and supported with adequate 'help' facilities. The transcript of a sample consultation follows, illustrating these and other features of the system.

In this example, the user sets out to find the value of the goal parameter, fire danger, in the Kay1 region which is one of the recognised land systems of Kakadu National Park. After this is determined to be "high", the user commences a second consultation in which a value for fire danger is sought in another region - the Queuel land system. Note that the user is not asked for the values of season and humidity a second time since season is equivalenced over all regions and humidity over both Kay1 and Queuel (see Figure 2). Note also the demonstration of the explanation facility of GEM.

All user input is in bold case.

```
****************************************************
*                                                  *
*   GEM An expert system for Geographic Domains    *
*                                                  *
*                        1985 Graham J. Williams   *
****************************************************
```

Name of Domain: **Kakadu**

EXECUTIVE CONTROL

EXECUTIVE> Type one of [consult,edit,trace,?]: **consult**

INTRO> Please type in your name: **Graham**

INTRO> What is the goal parameter? (Type ? for list) **?**

```
| Supply a Goal Parameter.                              |
|                                                       |
|    A goal parameter is required.  The choices         |
|    are:                                               |
|          available biomass:     [infer, ask]          |
|          grass biomass:         [infer, ask]          |
|          degree of curing:      [infer, ask]          |
|          fire danger:           [infer]               |
|          flame height:          [infer]               |
|          fuel type:             [geo, ask]            |
|          fuel weight:           [geo, ask]            |
|          grass flame:           [infer]               |
|          grass type:            [infer, ask]          |
|          litter flame:          [infer]               |
|          scorch height:         [infer]               |
|          description:           [geo]                 |
|                                                       |
---------------------------------------------------------
|                                                       |
|    Other commands available are:                      |
|      help, break, quit, halt                          |
|      pr (to print a rule)                             |
|      pp (to print a parameter)                        |
|      dump (to dump the knowledge base)                |
|      search (to search the geog. data file)           |
```

INTRO> What is the goal parameter? (Type ? for list) **fire danger**

INTRO> What land system are we to consider? (Type ? for list) **Kay 1**

What is the value of the season Graham?
The possibilities are:
cool, cold, hot, early storms, monsoon, knock em down storms]: **cool**

A value for the humidity is needed.
A percentage (0..100) is required: **30**

Tell me Graham what is the value of wind strength?
The possibilities are:
[calm, low, moderate, high]: **why**

WHY> The following rule was called upon
WHY> to determine a value for "fire danger".

Rule Number 27

If the season is cool, and
 the humidity is less than 40, and
 the wind strength is calm, and
Then
 the fire danger is low

Rule info :
 Certainty 100
 Author Jamie Hoare
 Regions [all]

From you I know that the season is cool (100%).

From you I know that the humidity is 30 (100%).
From rule 99 I know that the degree of curing is 100 (100%).

What is the value of the wind strength?
The possibilities are:
[calm, low, moderate, high]: **high**

>> It has been determined that
>> the fire danger is high in the Kay 1 region.

END OF CONSULTATION

END> Note: Previous answers kept unless wiped.
END> Type one of [consult,review,wipe,edit,trace,?]: **consult**

Are you Graham? [y or n] **y**

Stay with the parameter "fire danger" as the goal? [y or n] **y**

Stay with the land unit "Kay 1"? [y or n] **n**

INTRO> What land system are we to consider? (Type ? for list) **Queue 1**

What do you reckon the wind strength is Graham?
The possibilities are:
[calm, low, moderate, high]: **calm**

>> It has been determined that
>> the fire danger is low in the Queue 1 region.

END OF CONSULTATION

END> Note: Previous answers kept unless wiped.
END> Type one of [consult,review,wipe,edit,trace,?]: **pp**

Parameter Name: **season**

Parameter is "season".

```
Value Type            [cool, cold, hot, early storms,
                       monsoon, knock em down storms]
Regions Equivalenced  [all]
Source List           [ask]
Description           According to the Aboriginal calendar
```

END> Note: Previous answers kept unless wiped.
END> Type one of [consult,review,wipe,edit,trace,?]: **halt**

Are you sure you want to exit from this system? [y or n] **y**

Bye for now Graham.

5. The Implementation

GEM has been implemented using a standard PROLOG syntax (Clocksin and Mellish, 1981), on a DEC Rainbow computer. The code occupies approximately 100 KBytes, and the knowledge base with 100 rules, 3 geographic data items per region, and 17 parameters defined over 85 regions occupies 25 KBytes. As stated earlier, efficiency and speed were important considerations in the design of GEM, and the current version is successful in this respect. On rare occasions, the user waits up to 3 seconds between questions from the system, but usually questions occur 1 or 2 seconds apart. However, to ensure this level of interactivity, the authors used a small-core PROLOG system (128K) and consequently it has proven difficult to increase the knowledge base much beyond the current size.

Approximately three person-months were required to implement GEM. However, work is continuing on the development of the system.

The use of a widely accepted PROLOG standard has ensured that the system can easily be ported. The system has, for example, been ported to a SUN workstation, running MU-PROLOG (Naish, 1983), with only minor modifications.

A manual detailing how to use the system is available (Williams, 1985).

6. Suggestions for Improvement

The GEM shell has been used to develop a National Park expert system as outlined in this paper. However it has also been applied to other geographical problems and, from the experience gained, it is possible to envisage a number of improvements to the shell. Four are discussed here.

Firstly, as mentioned earlier, a complete implementation of the *possible-values* sub-system is not yet available. It is envisaged that the conclusions of the rules will be generalised to allow any relation to be specified in the Conclusion triplet, providing a much more flexible representation scheme. Problems associated with conflicts need to be addressed.

More generally, an analysis of many expert systems (Williams, 1986a) revealed that the issue of knowledge representation is a central factor in the shortcomings of expert systems. In GEM, too, difficulties arose with the inability to represent hierarchical information, and the inability to cleanly associate knowledge with those regions in which it is applicable. Such difficulties can be traced to the use of production rule and database knowledge representations. In order to address such difficulties a frames formalism (Greiner and Lenat, 1980) has been explored (Williams, 1986b). The details of this knowledge representation will not be described here. Advantages from using such a scheme though result from the hierarchical nature of frames, and the flexibility they offer in representing information about how to compute values for parameters.

As mentioned earlier, an algorithmic base is to be implemented to allow the use of arithmetic procedures (essentially equations). These will encode the sort of knowledge typically built up by scientists (Davis et. al., 1985). Whilst many expert systems do not facilitate the representation of this knowledge, concentrating more on heuristic knowledge, in the present case some procedural information was available, and it was felt wise to make use of this using a procedural rather than declarative representation.

Over the past decade it has become clear that for expert systems to be effective they require a large store of knowledge (Feigenbaum, 1977). Typically, this knowledge takes months to encode (Buchanan, 1982); for example, Kakadu knowledge base took 4 months of 2-3 days per week involving two persons, and is still evolving. Much current research in the field of Artificial Intelligence in general, and expert systems in particular, is focussing on this problem, and a number of approaches have been proposed (Nilsson, 1980), including intelligent editors and learning systems.

The authors are undertaking research to apply learning systems to geographic databases in order to obtain frames which will be used in a geographic expert system. They are also developing an intelligent editor which will assist the expert explicate his/her knowledge.

7. Summary

This paper reports the first stage of the development of the GEM expert system shell for classification problems in the geographic domain. It possesses features for restricting knowledge to certain regions of a geographic area, a simple means of transferring values of parameters between regions, and three methods of representing knowledge/information. At its present state of development it has been found to be sufficiently complete to be useful in a number of applications, notably estimating the effect of fires in a major National Park. It is expected that as these applications develop the features and capabilities of the shell program will be extended; in the meantime, four areas of improvement have been identified mainly from the National Park application: two requiring extensions to the knowledge representation, operationalising the *possible-values* concept incorporated in the present version, and developing computerised aids for knowledge acquisition.

Availability

The GEM shell can be obtained from the CSIRO, Division of Water and Land Resources, G.P.O. Box 1666, Canberra, Australia 2601.

Acknowledgements

Thanks go to Jamie Hoare of the Division of Forest Research, CSIRO, without whose knowledge, and indeed willing assistance, we would have no knowledge base.

References

Australian National Parks and Wildlife Service. *Kakadu National Park Plan of Management* (Report). Australian National Parks and Wildlife Service, 1980.

Buchanan, B.G. New Research on Expert Systems. In J.E. Hayes, D. Michie, and Y-H. Pao (Eds.), *Machine Intelligence*. Vol. 10: *Intelligent Systems: Practice and Perspective*. Chichester: Ellis Horwood, 1982.

Clocksin, W.F., and Mellish, C.S. *Programming in Prolog*. Springer-Verlag, 1981.

Cuena, J. The use of simulation models and human advice to build an expert system for the defense and control of river floods. In *Proceedings of the Eigth International Joint Conference on Artificial Intelligence*. Karlsruhe: IJCAI-83, 1983.

Davis, R. and King J.J. The origin of rule-based systems in AI. In B.G. Buchanan and E.H. Shortliffe (Eds.), *Rule-based expert systems*. Reading, Massachusetts: Addison-Wesley, 1984.

Davis, J.R., and Nanninga, P.M. GEOMYCIN: Toward a Geographic Expert System for Resource Management. *Journal of Environmental Management*, 1985, 21, 377-390.

Davis, J.R., Nanninga, P.M., and Cocks, K.D. The Usefulness of Computer Aids that Capture Expert Knowledge about Land Management. In *Proceedings of the First*

International Conference on Rural Land Management:. University of Guelph, Canada, 1985.

Davis, J.R., Hoare, J.R.L., and Nanninga, P.M. Developing a Fire Management Expert System for Kakadu National Park, Australia. *Accepted by Journal of Environmental Management*, 1986, Vol. 22.

Feigenbaum, E.A. The Art of Artificial Intelligence: 1. Themes and case studies of knowledge engineering. In *Proceedings of the Fifth International Joint Conference on Artificial Intelligence.* MIT, Cambridge, Massachusetts: IJCAI-77, 1977.

Greiner, R., and Lenat, D.B. A Representation Language Language. In *Proceedings of the First National Conference on Artificial Intelligence.* Stanford, Califonia: AAAI-80, 1980.

Hayes-Roth, F., Waterman, D.A., and Lenat, D.B., eds. *Teknowledge Series in Knowledge Engineering.* Vol. 1: *Building Expert Systems.* Reading, Massachusetts: Addison-Wesley, 1983.

Hoare, J.R.L. A model of fire behaviour and resulting damage to vegetation for predicting the biological effects of fire on plant communities at Kakadu National Park. In *Towards an Expert System for Fire Management at Kakadu National Park.* Canberra: Technical Memorandum 85/2, CSIRO Division of Water and Land Resources, 1985.

Mackenzie, H.G. *EMYCIN users guide.* (Technical Report 17). Division of Computing Research, CSIRO, Canberra, 1984.

McDermott, D. and Davis, E. Planning routes through Uncertain Territory. *Artificial Intelligence*, 1984, 22, 107-156.

Naish, L. *MU-PROLOG 3.0 Reference Manual* (Technical Report). Dept. Computer Science. University of Melbourne, July 1983.

Newell, A., and Simon, H.A. *Human Problem Solving.* Englewood Cliffs, NJ: Prentice-Hall, 1972.

Nilsson, N.J. *Principles of Artificial Intelligence.* Palo Alto, California: Tioga, 1980.

Pereira, L.M., Sabatier, P. and de Oliveira, E. *ORBI - An Expert System for Environmental Resource Evaluation through Natural Language* (Tech. Rep. 3/82). Dept. de Informatica, Universidade Nova de Lisboa, 1982.

Shortliffe, E.H. *Artificial Intelligence Series.* Vol. 2: *Computer-Based Medical Consultations: MYCIN.* New York: American Elsevier, 1976.

Smith, T.R. Artificial Intelligence and its applicability to geographic problem solving. *Professional Geographer*, 1984, 36, 147-158.

Starfield, A.M., Owen-Smith, N., and Bleloch, A.L. A rule-based population model for adaptive management. *South African Journal of Wildlife Research*, 1985, 15(2), 59-62.

Story, R., Williams, M.A.J., Hooper, A.D.L., O'Ferrall, R.E., and McAlpine, J.R. *Lands of the Adelaide-Alligator Area, Northern Territory* (Land Research series 25). CSIRO, Melbourne, Australia., 1969.

Story, R., Galloway, R.W., McAlpine, J.R. , Aldrick, J.M. and Williams, M.A.J. *Lands of the Alligator Rivers Area, Northern Territory* (Land Research series 38). CSIRO, Melbourne, Australia., 1976.

Walker, J., Davis, J.R., and Gill, A.M. *Towards an expert system for fire management at Kakadu National Park* (Technical Memorandum 85/2). Division of Water and Land Resources, CSIRO, Canberra, 1985.

Waterman, D.A. *A Guide to Expert Systems*. Reading, Massachusetts: Addison-Wesley, 1986.

Weiss, S.M., and C.A. Kulikowski. *A Practical Guide to Designing Expert Systems*. New Jersey: Rowman & Allanheld, 1984.

Williams, G.J. *GEM Users' Manual* (Technical Memorandum 85/16). Division of Water and Land Resources, CSIRO, Canberra, October 1985.

Williams, G.J. Deficiencies of Expert Systems and Attempts at Improvement: A survey. In *Proceedings of the Ninth Australian Computer Science Conference*. Dept. Computer Science, Australian National University, Canberra: ACSC9, 1986.

Williams, G.J. *FrameUP: A Frames Formalism for Expert Systems* (Technical Report). Department of Computer Science, Australian National University, 1986.