A Production System Model for Terrain Analysis Knowledge Representation

DEMETRE P. ARGIALAS and RAVI NARASIMHAN[1]
Louisiana State University

A production system model has been developed for terrain analysis problem solving. The working memory of the production system is used to store specific domain knowledge about landforms. This involved the description of landforms in terms of their pattern elements, including their likelihood of occurrence. The production memory was used to store the rules of inferencing. These rules were general rules and were applicable to all landforms. Thus, the domain knowledge about terrain analysis was separated into two components: one component consisting of specific knowledge about landforms, stored as facts in the working memory; and the other component consisting of the general methodology for inferencing, stored as rules in the production memory. Such a separation of knowledge enables additions to the knowledge base fairly easily. The knowledge base can be extended to encompass more landforms simply by creating more working-memory elements. The present version of the prototype expert system has been implemented using OPS5, a production system language. Uncertainty calculations were performed by invoking LISP functions from OPS5. The results indicated that the production system model was appropriate for designing the prototype expert system for terrain analysis.

INTRODUCTION

Terrain analysis is the systematic study of image patterns relating to the origin, morphologic history, and composition of distinct terrain units, called landforms [21, 26]. Terrain analysis takes into account and provides information about physical site factors such as geologic type and structure, soil type and its properties, vegetation type, drainage pattern type, and others. This information is used by engineers and planners for site development and identifying areas which require ground investigations such as borings and other types of field surveys.

Among the various approaches to terrain analysis, the landform-pattern element approach has been more prominent in the U.S.A. [18, 21, 26]. The landform-pattern element approach is based on the premise that any two landforms derived from the same soil and bedrock, or deposited by a similar process, and existing under the same climatic conditions, exhibit similar physical and visual features on aerial images, called "pattern elements" [21]. Terrain analysts use the pattern elements to identify and analyze landforms and to evaluate their engineering significance.

use

The pattern elements examined in the landform-pattern element approach include topographic form, drainage pattern type and texture, gully characteristics, soil tone, landuse type, and other special features that may be present. Typical descriptions of the topography of a landform are steep slopes; medium slopes; undulating, massive hills; table rocks; and others. Drainage pattern texture is usually classified as fine, medium, or coarse. The main drainage pattern types are dendritic, pinnate, parallel, rectangular, angular, trellis, radial, annular, and internal. Soil tones are described as white, light gray, medium gray, and dark. Landuse is classified as cultivated, forested, urban, and others. Gully types are distinguished as V-shaped, U-shaped, and sag-and-swale.

Although progress has been made toward the computational identification of certain pattern elements [2, 3], limited computational approaches have been developed to model terrain analysis logic, that is, the prob-

---

Address correspondence to: Demetre P. Argialas, Assistant Professor[1] and Graduate Student,[2] Department of Civil Engineering, Louisiana State University, Baton Rouge, LA 70803.

Microcomputers in Civil Engineering **3**, 55–73 (1988)

0885-9507/88/$3.50

lem solving strategy of expert terrain analysts [21]. Leighty [16] employed a logical approach for terrain pattern recognition and he [17] has suggested the use of rule-based systems for terrain analysis problem solving.

Advancements in artificial intelligence research and the subsequent emergence of expert systems have provided a new powerful tool for the development of computer programs that can capture expertise in many fields and tasks [12, 27]. Applications of expert systems in civil engineering have been presented in Kostem and Maher [15], Adeli [1], and Maher [19]. Expert systems have been successfully employed for representation of knowledge related to interpretation tasks, including urban scene interpretation [20], site evaluations for mineral resources [9], and military intelligence [8, 11].

The same approach needs to be pursued for computational modeling of the terrain analysis problem-solving process if substantial progress is to be made at modeling photointerpretation logic and at extracting terrain-related features automatically from aerial images. In this effort, an expert system approach was undertaken because terrain analysis problem solving requires knowledge that is largely empirical, heuristic, partial and incomplete, and computer representation of such knowledge cannot easily be held to rigid and exact descriptions available through procedural languages. Instead, it is greatly facilitated by symbolic representation, symbolic logic, and heuristic search. Furthermore, the interpretation of the landform of a site is a process not easily amenable to rigorous and complete modeling. Instead, uncertainties are introduced during problem solving in both the identification of the individual pattern elements and the synthesis of the pattern elements in inferring the landform of a site [4]. Hence, expert systems can be an ideal environment for studying terrain analysis methodologies and learning more about this process.

The overall objective of this research effort was the development of knowledge representation and inference schemes for terrain analysis problem solving and the implementation of these schemes in a rule-based production system language. The result was a Terrain Analysis Expert (TAX) system prototype.

The goal of a typical consulting session with the Terrain Analysis Expert (TAX) system was to infer the landform type of a site given a stereopair of aerial photographs. The approach followed for inferring the landform of the site was the landform-pattern element approach [21]. Knowledge pertaining to the landform-pattern element approach was represented in physiographic section models and landform models [4]. The physiographic section models represented the relations among sections and landforms located in them. The landform models contained information about all the pattern element values that were likely to be found in a landform, and the likelihood of their occurrence.

KNOWLEDGE-BASED EXPERT SYSTEMS

Since expertise in a task domain requires substantial knowledge about that domain, effective representation of domain knowledge is considered to be a key factor that determines the success of an expert system. In current expert system paradigms, this knowledge is represented as a combination of specific domain knowledge (expert's knowledge) and general problem-solving knowledge, that is, the knowledge about how to make effective use of the domain knowledge. This is usually achieved by developing a knowledge framework as a combination of representation, inference, and control [7, 12–14, 27].

There are a number of knowledge representation, inference, and control schemes, any of which can be used alone or in conjunction with others to build expert systems. Each scheme can provide an expert system with certain benefits, such as making it more efficient, more easily understood, or more easily modified. The best developed knowledge representation schemes employed in current expert systems are production rules, predicate calculus, semantic networks, and frames [14]. Inference techniques employed in current expert systems include modus ponens, resolution, reasoning about uncertainty, and others [12]. Control schemes such as forward and backward chaining, forward and backward reasoning, depth first and breadth first search can be employed for controlling attention during problem solving [13].

Expert systems are usually employed in domains where facts, rules, and, consequently, conclusions are rarely certain or exact. Inexact reasoning procedures have, therefore, been developed to complement the knowledge representation and inferencing mechanisms of rule-based systems. For example, designers often build some sort of certainty computing procedure on top of the basic antecedent-consequent mechanism.

Generally, certainty computing procedures associate a number between, say, -3 to $+3$, with each domain fact. This number, called a certainty factor, is intended to reflect how certain the fact is, with -3 indicating a fact is definitely false and $+3$ indicating a fact is definitely true. Some of the established procedures for handling inexact reasoning, that have been demonstrated in the well-known expert systems, such as MYCIN [6, 25], PROSPECTOR [9, 23], and HYDRO [24], employ heuristic techniques, which are approximations of Bayes' theorem, for handling probabilities and certainties. These heuristic techniques provide a way for representing uncertainties in facts, in combination of facts, in inference rules, and in facts supported independently by several rules [6, 7, 23, 25, 27].

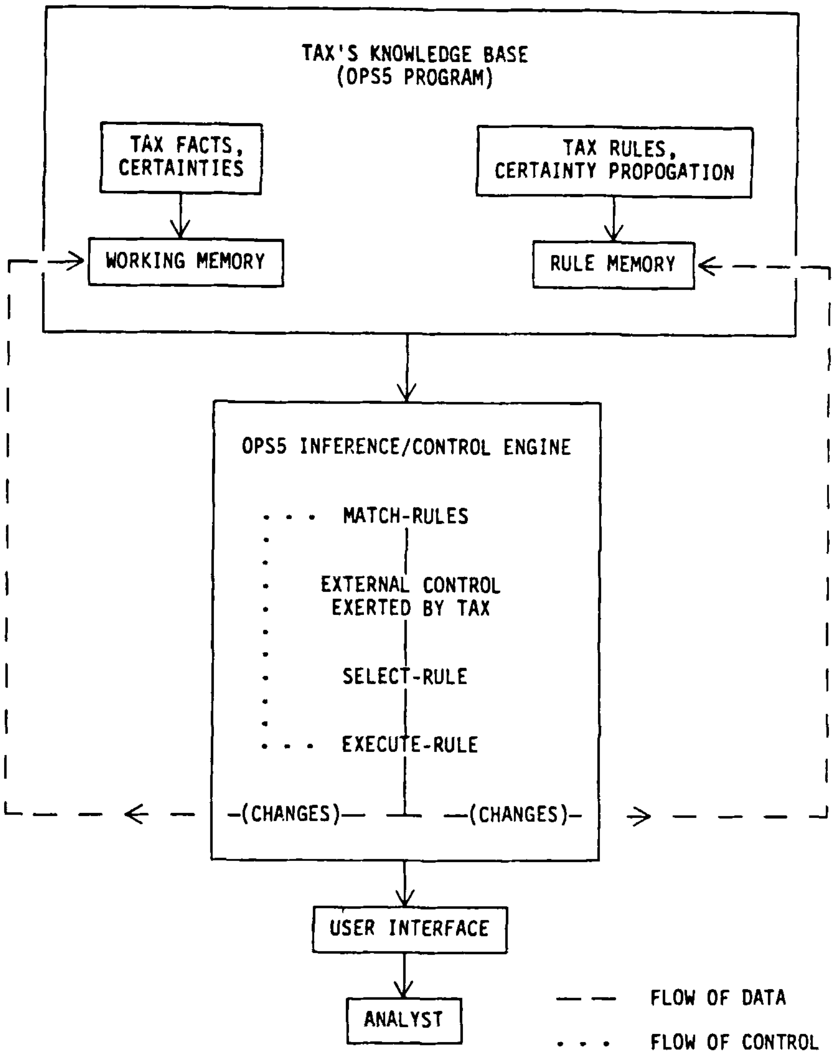

FIGURE 1. Architecture of TAX's production system model.

DESIGN OF TAX

The methodology in building the Terrain Analysis Expert (TAX) system involved design of knowledge representation and inference schemes and their implementation in an expert system tool. Since there was no previous work published on employing expert systems for terrain analysis, the choice was made for the rather well established "rule-based" approach. Among alternatives in "rule-based" systems, a production system model was selected. In particular, an initial commmitment was made to the production system architecture of the OPS5 language [10]. OPS5 was chosen because 1) it is widely available for minicomputers, workstations and microcomputers; 2) it is well documented and discussed in-depth in Forgy [10], and Brownston et al. [5]; 3) it has been used for a commercial large-scale system XCON (or R1) [12]; and 4) it offers a flexible forward-chaining control strategy, while providing for external control, by system designers, through production rules.

The key TAX concepts, objects, and decision rules were represented in a formal way, within the representation, inference, and control schemes of the OPS5 language. The TAX's production system model consisted of a knowledge base and an inference engine (Figure 1). The knowledge base was composed of facts and rules and it constituted the OPS5 program code.

KNOWLEDGE REPRESENTATION SCHEME

Terrain analysis facts are represented as working memory elements and grouped as element classes or objects. Such objects are declared by the "literalize" command and the working memory elements are created by the "make" command [5, 10]. For example, the following literalize command contains the description of a landform, in the form of the object PATTERN-ELEMENTS-OF-LANDFORM

```
(literalize pattern-elements-of-landform
landform-name
topography
drainage-texture
drainage-pattern-type
gully-type
gully-amount
soil-tone
land-use-hilltops
land-use-valleys)
```

and the following "make" command creates the class element for humid sandstone (the symbol ^ indicates that what follows is an attribute name)

```
(make pattern-elements-of-landform
^landform-name            humid-sandstone
^topography               steep-slopes
^drainage-texture         coarse
^drainage-pattern-type    dendritic
^gully-type               v-shaped
^gully-amount             few
^soil-tone                light-gray
^land-use-hilltops        forested
^land-use-valleys         agriculture)
```

However, each pattern element of a landform could have multiple values, i.e., the topography could be steep slopes or medium slopes. OPS5 does not permit an element class such as PATTERN-ELEMENTS-OF-LANDFORM to store multiple values of attributes. This problem can be overcome by declaring the attributes to be vector attributes. Unfortuantely, in OPS5, only one attribute in an element class can be declared as a vector attribute [5, 10].

This representation problem was solved by creating an object (element class) for each pattern element, such as

```
(literalize landform-topography-pair
name-of-landform
topography)
(literalize landform-topography-pair
name-of-landform
drainage-type)
```

Multiple values of pattern elements were accommodated, by making multiple elements having the same name-of-landform attribute, but with different values of the pattern element, such as

```
(make landform-topography-pair
^name-of-landform    sandstone-humid
^topography          medium-slopes
(make landform-topography-pair)
^name-of-landform    sandstone-humid
^topography          steep-slopes
```

Since OPS5 does not embody uncertainty handling techniques, it was decided to implement such a mechanism on top of OPS5 [4]. The methodology introduced accounted for the uncertainties, in the observation of pattern elements and their role in the establishment of the landform of the site by employing probabilities in the models of landforms, such as

```
(make landform-topography-pair
^name-of-landform    sandstone-humid
^topography          medium-slopes
^topography-P(E/H)   0.2)
(make landform-topography-pair
^name-of-landform    sandstone-humid
^topography          steep-slopes
^topography-P(E/H)   0.6)
```

The attribute topography-P(E/H) represented probability values in a manner similar to that employed in PROSPECTOR [9, 23]. P(E/H) was the probability of the occurrence of the pattern element value in that landform, or probability of the evidence given the hypothesis. The values of P(E/H) were initially extracted from books and reports [18, 21, 26]. They were later refined by the expert [22]. External FRANZLISP [5] functions were written to update the probabilities.

When the knowledge base was created, the elements corresponding to each landform-pattern-element-pair class were arranged in the ascending order of the likelihood of occurrence of the pattern element, such as

```
(make landform-drainage-type-pair
^name-of-landform             sandstone-humid
^drainage-type                angular
^drainage-type-probability    0.2)
(make landform-drainage-type-pair
^name-of-landform             sandstone-humid
^drainage-type                rectangular
^drainage-type-probability    0.2)
(make landform-drainage-type-pair
^name-of-landform             sandstone-humid
^drainage-type                dendritic
^drainage-type-probability    0.6)
```

This ordering ensured that the pattern element which was most likely to occur was the most recently created (e.g., last) in its element class. Consequently, a rule such as site drainage type, which queried the analyst for a certainty value for the drainage type, started with the most likely pattern element, i.e., dendritic drainage. If the user gave a high certainty value (≥ 2), then no more questions were asked about the drainage type of the site. This improved the efficiency of the program by exploring paths in the order of likelihood of achieving success.

Besides the landform-pattern-elements objects, other type of objects were designed. The models describing the relationship between physiographic sections and landforms were similarly represented as shown in Table 1, with the object SECTION-LANDFORM-PAIR with attributes section-name, landform-type, and the probability representing the likelihood of occurrence of that

TABLE 1 SAMPLE OF TAX'S OBJECTS REPRESENTED ON OPS5

LANDFORM_TOPOGRAPHY_PAIR	
^landform_type	<landform-value>
^topography	<topography-value>
^landform_topography_peh	<peh-value>
^landform_topography_penoth	<penoth-value>
^status	nil
SECTION_LANDFORM_PAIR	
^section_name	<section-value>
^landform_type	<landform-value>
^section_landform_prob	<probability-value>
LANDFORM_OF_THE_SITE	
^landform_type	<landform-value>
^probability	<topography-value>
^status	nil
TOPOGRAPHY_OF_THE_SITE	
^landform_type	<landform-value>
^topography	<topography-value>
^certainty_value_of_topography	<certainty-value>
^status	nil

landform in the corresponding physiographic section. Table 2 shows plausible values assigned to represent the occurrence of humid limestone in the Cumberland plateau section.

The object LANDFORM-OF-THE-SITE was designed for representing the landform of the site with two attributes: landform-type and its associated probability (Tables 1 and 2). The object TOPOGRAPHY-OF-THE-SITE was constructed to represent the topography of the site by providing the association between the hypothesized landform type of the site, the value of the topography of the site, and the certainty of the value of topography (Tables 1 and 2). Additional objects were defined to describe the rest of the pattern elements and other required objects.

INFERENCE SCHEME

The problem-solving strategy was represented through production rules pertaining to the preceding defined objects. Production rules, written in OPS5, were represented by condition-action pairs [5, 10]. Each rule had a condition part which was composed of a conjunction of one or more antecedent clauses, and an action part, the consequent, which created or modified working-memory elements. Both the antecedent and consequent parts were logical combination of clauses, the antecedent part specifying the preconditions, and the consequent part specifying a set of actions modifying the working memory by adding, deleting, or changing facts.

The purpose of some of the rule clauses was to match the site's pattern elements to the landform models stored in the working memory. Other clauses affected the interrelationships and interactions among the rules, and so they exerted control on the execution of the rules over and beyond that imposed by the inference engine of OPS5.

TAX employs backward reasoning to identify a landform. At first, the a priori certainty associated with the hypothesis of a landform is estimated from information related to the physiography of the site. Rule hypothesize-a-landform-type-based-on-physiography (Table 3) creates working-memory elements of the type SECTION-LANDFORM-PAIR representing the hypothesized landforms that can occur in that physiographic section. The a priori certainty of each hypothesized landform is initialized to the probability of the occurrence of the landform in that physiographic section. TAX then selects these hypothesized landforms, one by one, and attempts to

TABLE 2 PLAUSIBLE VALUES OF THE OBJECTIVE ATTRIBUTES OF TABLE 1 FOR HUMID LIMESTONES

```
LANDFORM_TOPOGRAPHY_PAIR

     ^landform_type                          sandstone_humid
     ^topography                             steep_slopes
     ^landform_topography-peh                0.60
     ^landform_topography_penoth             0
     ^status                                 nil

SECTION_LANDFORM PAIR

     ^section_name                           cumberland_plateau
     ^landform_type                          limestone-humid
     ^section_landform_prob                  0.1

LANDFORM_OF_THE_SITE

     ^landform_type                          sandstone_humid
     ^probability                            0.45
     ^status                                 nil

TOPOGRAPHY_OF_THE_SITE

     ^landform_type                          sandstone_humid
     ^topography                             steep_slopes
     ^certainty_value_of_topography          +1
     ^status                                 nil
```

TABLE 3 RULE HYPOTHESIZE-A-LANDFORM TYPE-BASED-ON-PHYSIOGRAPHY IN OPS5 LANGUAGE

```
(P hypothesize_a_landform_type_based on_physiography

     (section_landform_pair

          ^section_name                 <section_value>
          ^landform_type                <landform_value>
          ^section_landform_prob        <probability_value>)
-->
     (make landform_of_the_site

          ^landform_type                <landform_value>
          ^probability                  <probability-value>

     )

)
```

TABLE 4 RULE QUERY_SITE_TOPOGRAPHY_FROM_ANALYST IN OPS5 LANGUAGE

```
(p Query_site_topography_from_analyst
       {(landform-of-the-site
              ^landform-type  <landform-value>
              ^status         nil
              )
         <site-landform>}
       {(landform-topography-pair
              ^landform-type  <landform-value>
              ^topography     <topography-value>
              ^status         nil
              )
         <landform-topography>}
       - (topography-of-the-site
              ^topography     <topography-value>
              )
       - (topography-of-the-site
              ^landform-type  <landform-value>
              ^certainty-value-of-topography  >=      2)
→
       (write (crlf) Is the topography of the site (crlf)
                    <topography-value> ?
              Give a certainty value between -3 to +3
              (crlf))
       (make topography-of-the-site
              ^landform-type  <landform-value>
              ^topography     <topography-value>
              ^certainty-value-of-topography  (accept))
       (modify <site-landform>
              ^status         nil)
       (modify <landform-topography>
              ^status         done
              )
)
```

establish each one of them by matching the pattern elements of the site with the models of the landforms [4].

The matching of the pattern elements of the site to the pattern elements of each of the hypothesized landforms takes place by first querying the analyst for a certainty value (between -3 and $+3$) for each pattern element. Rules such as query-site-topography-from-analyst (Table 4) perform the task of obtaining certainty values of the pattern elements of the site. If the pattern element value has already been questioned, while establishing another landform, then rules like infer-site-topography-if-already-there (Table 5) are used to infer the certainty values to support the current hypothesis.

If the model of a landform contains multiple values for a pattern element, the TAX queries the analyst for all the values or until a certainty of 2 or more is given by the analyst for a particular pattern element value. At this stage, rules like establish-site-topography-type (Table 6) are fired which select the best pattern element value. For the pattern element value with the highest certainty value, the $P(E/^{-}H)$ value is computed by employing rules like compute-site-topography-$P(E/^{-}H)$ (Table 7). Finally, rules like update-hypothesis-based-on-site-topography (Table 8), pertaining to all pattern elements, are fired to modify the a priori certainty associated with the hypothesis of the landform.

This matching procedure is repeated for all pattern elements and for all hypothesized landforms. The landform which has the highest a posteriori certainty associated with it is declared to be the landform of the site by firing of rule display-conclusions (Table 9).

The OPS5 inference engine provides for matching of rules, selecting rules, and executing rules [5, 10]. First, it determines which rule instantiations are relevant to a

TABLE 5 RULE INFER_SITE_TOPOGRAPHY_IF_ALREADY_THERE IN OPS5 LANGUAGE

```
(p Infer_site_topography_if_already_there
        {(landform-of-the-site
                ^landform-type  <landform-value>
                ^status         nil
                )
          <site-landform>}
        {(landform-topography-pair
                ^landform-type  <landform-value>
                ^topography     <topography-value>
                ^status         nil
                )
          <landform-topography>}
         (topography-of-the-site
                ^topography     <topography-value>
                ^landform-type
                 {<other-landform-value> <> <landform-value>}
                ^certainty-value-of-topography  <cert>
                )
        - (topography-of-the-site
                ^landform-type  <landform-value>
                ^certainty-value-of-topography  >=      2)
        - (topography-of-the-site
                ^landform-type  <landform-value>
                ^topography     <topography-value>
           )
->
        (make topography-of-the-site
                ^landform-type  <landform-value>
                ^topography     <topography-value>
                ^certainty-value-of-topography  <cert>
        )
        (modify <site-landform>
                ^status         nil)
        (modify <landform-topography>
                ^status         done
                )
)
```

given working-memory configuration and assembles them in a conflict set. Then, it uses a conflict resolution strategy to fire one rule from the conflict set.

OPS5 allows a choice of two conflict resolution strategies: LEX which stands for lexicographic ordering, and MEA which stands for means ends analysis [5, 10]. The difference between the two strategies is that MEA places extra emphasis on the recency tag of the working-memory element matching the first condition element in a rule. The first step in both strategies is refraction, which removes from the conflict set all instantiations which have been already fired. In the MEA strategy, this step is followed by a comparison of the instantiations based on the recency of the first condition element. The instantiation which is the most recent is selected. If no single instantiation dominates, the resultant set is passed through the subsequent steps of conflict resolution which are similar in both the LEX and MEA strategies.

TAX employs the MEA strategy for resolving conflicts (Appendix I). The advantage of using MEA is that by a proper ordering of the condition elements and subtle modifications of the recency tags of the working-memory elements, the desired order of rule firings is realized. For instance, in the right-hand side of the rule query-site-topography-from-analyst (Table 4), the attribute *status* of the element LANDFORM-OF-THE-SITE is reset to

TABLE 6 RULE ESTABLISH_SITE_TOPOGRAPHY_TYPE IN OPS5 LANGUAGE

```
(p Establish_site_Topography_type
        {(topography-of-the-site
                ^landform-type  <landform-value>
                ^topography     <landform-topography-match>
                ^certainty-value-of-topography <certainty-value>
                ^status         nil)
           <site-topography>}
        - (topography-of-the-site
                ^landform-type  <landform-value>
                ^topography     <topography-value>
                ^certainty-value-of-topography
                    { > <certainty-value>} )
        - (landform-topography-pair
                ^landform-type  <landform-value>
                ^status         final
          )
        {(landform-topography-pair
                ^landform-type  <landform-value>
                ^topography     <landform-topography-match>
                ^status         done
          )
                <landform-topography>}
→
        (modify <landform-topography>
                ^status         final
          )
        (modify <site-topography>
                ^status         done
          )
)
```

nil. This ensures that this element is the most recent, and also ensures repeated firing of the same rule, with different instantiations, until all possible instantiations are exhausted. Only after all the possible values have been queried or inferred, the rule establish-site-topography-type (Table 6), which establishes the "best topography," is fired. Similarly, in the rule display-conclusions (Table 9), the first condition element is the element SITE, which is one of the first working-memory elements to be created. The inclusion of such an element, as the first condition element, ensures that the rule display-conclusions is the last to fire. In all conflict sets, this rule is the first to be discarded as long as there are other eligible rule instantiations.

As an example of the structure and semantics of OPS5 rules, the English version of two OPS5 rules query-site-topography-type (Table 4) and establish-topography-of-the-site (Table 6) are given in Tables 10 and 11, respectively.

Facts are stored in the working memory which serves as a global data base of symbols representing facts and assertions about the terrain-related objects. The working memory of OPS5 is a dynamic data base of facts and consequently, during program execution, it is continually modified by the action part of the production rules. Table 12 shows an example of working-memory configuration before and after the firing of rule establish-site-topography-type. The effect of this rule firing is to modify the attribute values of two working-memory elements: TOPOGRAPHY-OF-THE-SITE and LANDFORM-TOPOGRAPHY-PAIR. Similar modifications of the working memory results from firings of other rules.

TESTING AND EVALUATION

Designing and implementing TAX involved constant evaluation of the progress by considering such aspects as 1) the adequacy of the representation and inference schemes, 2) the consistency and accuracy of the embedded knowledge with that of experts, and 3) the accuracy and correctness of the conclusions the system provided.

TABLE 7 RULE COMPUTE_SITE_TOPOGRAPHY_P(E/¯H) IN OPS5 LANGUAGE

```
(p Compute_site_topography_P(E/¯H)
        {(landform-topography-pair
                ^landform-type  <landform-value>
                ^topography     <topography-value>
                ^landform-topography-penoth-num <penoth-num>
                ^landform-topography-penoth-den <penoth-den>
        ) <landform-topography>}
        (topography-of-the-site
                ^landform-type  <landform-value>
                ^topography     <topography-value>
                ^status         done
        )
        (landform-topography-pair
                ^landform-type
                  { <other-landform-value> <> <landform-value> }
                ^topography     <topography-value>
                ^landform-topography-peh                <peh>
        )
        (section-landform
                ^section        <phys-secn>
                ^landform-type  <other-landform-value>
                ^section-landform-prob  <prob>
        )
        - (penoth-topography
                ^topography     <topography-value>
                ^landform-type  <landform-value>
                ^other-landform <other-landform-value>
        )
→
        (modify <landform-topography>
                ^landform-topography-penoth-num
                        (compute _penoth-num_ +
                        <peh> * <prob> )
                ^landform-topography-penoth-den
                        (compute <penoth-den> +
                         <prob> )
        )
        (make penoth-topography
                ^topography     <topography-value>
                ^landform-type  <landform-value>
                ^other-landform-type    <other-landform-value>
        )
)
```

The adequacy of the conceptual representation of the problem and of the production model formulation were tested by implementing the system in OPS5. Implementation was itself a form of testing. In this sense, testing involved the evaluation of the performance and utility of the early prototype program and some revisions as were found necessary. Problems with the representation scheme, such as missing objects, facts, and inappropriate certainty values and rules were identified. Such problems have forced reformulating concepts, and redesigning the production rules so as to provide for a more rational utilization of the expert's knowledge and a more efficient assessment of the landform of the site through the production system model.

TABLE 8 RULE UPDATE_HYPOTHESIS_BASED_ON_SITE_TOPOGRAPHY IN OPS5 LANGUAGE

```
(p Update_Hypothesis_based_on_site_Topography
    {(landform-of-the-site
        ^landform-type  <landform-value>
        ^certainty-of-hypothesis <certainty>
        ^status         nil
    ) <site-landform>
    {(topography-of-the-site
        ^landform-type  <landform-value>
        ^topography     <topography-value>
        ^certainty-value-of-topography <certainty-value>
        ^status         done
    ) <site-topography>}
    (landform-topography-pair
        ^landform-type  <landform-value>
        ^topography     <topography-value>
        ^landform-topography-peh  <peh>
        ^landform-topography-penoth-num  <penoth-num>
        ^landform-topography-penoth-den <penoth-den>
    )
→
    (modify <site-landform>
        ^certainty-of-hypothesis (certainty_update
            <landform-value> <certainty>
               <certainty-value>
            <peh> <penoth-num> <penoth-den>)
    )
    (modify <site-topography>
        ^status         final
    )
)
```

TABLE 9 RULE DISPLAY-CONCLUSIONS IN OPS5 LANGUAGE

```
(p display-conclusions
         (site)
        {(landform-of-the-site
                 ^landform-type  <best-landform>
                 ^certainty-of-hypothesis <best-certainty>)
         <site-landform-match>}
        - (landform-of-the-site
                 ^status         final)
        - (landform-of-the-site
                 ^certainty-of-hypothesis
                      { > <best-certainty>})
→
        (modify <site-landform-match>
                 ^status         final)
        (write (crlf) (crlf) The site appears to be
        ******  <best-landform>  ******
         )
        (call trunc landform-of-the-site
                         ^landform-type  <best-landform>
                         ^certainty-of-hypothesis
                                        <best-certainty>
                )
)
```

TABLE 10 ENGLISH VERSION OF RULE QUERY_SITE_TOPOGRAPHY_FROM_ANALYST

IF	the attribute landform_type has not as yet been established for the LANDORM_OF_THE_SITE
and	there is a LANDFORM_TOPOGRAPHY_PAIR in the KB, whose attribute landform_type is "landform_value" and its attribute topography is "topography_value" (and this "topography_value" has not as yet been tested as being a plausible value of the topography of the LANDFORM_OF_THE_SITE)
and	there is no TOPOGRAPHY_OF_THE_SITE whose attribute topography has been querried
and	there is no TOPOGRAPHY_OF_THE_SITE with attribute landform_type being "landform_value", whose attribute certainty_value_of_topography, obtained from the analyst, is greater or equal to 2
THEN	obtain, by querying the analyst, a "certainty-value" (between -3 and +3) for the value "topography-value" of the attribute topography of the LANDFORM_TOPOGRAPHY_ELEMENT
and	create TOPOGRAPHY_OF_THE_SITE with attribute landform_type set to "landform_value" and attribute topography set to "topography_value" and attribute certainty_value_of_topography set to the obtained "certainty_value"
and	modify the attribute status for the LANDFORM_OF_THE_SITE to NIL, to ensure that the TOPOGRAPHY_OF_THE_SITE will be reevaluated
and	modify the attribute status of the LANDFORM_TOPOGRAPHY_PAIR to "done", indicating that the value of the attribute topography, "topography_value", which corresponds to attribute landform_type (of the KB), with value "landform-value", has been tested as being a plausible value of the TOPOGRAPHY_OF_THE_SITE

TABLE 11 ENGLISH VERSION OF RULE ESTABLISH_SITE_TOPOGRAPHY_TYPE

IF	there is a TOPOGRAPHY_OF_THE_SITE element, pertainig to certain landform_type, say "landform_value", whose attribute topography has the value "landform_topography_match" and the certainty_value_of_topography is "certainty_value"
and	there is no TOPOGRAPHY_OF_THE_SITE element whose certainty_value_of_topography is greater than "certainty_value", for the landform_type "landform_value"
and	there is no LANDFORM_TOPOGRAPHY_PAIR element for which final evaluation was made as being the best
and	there is a LANDFORM_TOPOGRAPHY_PAIR element, pertaining to "landform_value", whose topography value is "topography-value"
THEN	modify the LANDFORM_TOPOGRAPHY_PAIR element to indicate that a best and final value evaluation was made favoring the landform_type with the highest certainty_value_of_topography

TABLE 12 PARTIAL WORKING MEMORY CONFIGURATION BEFORE AND AFTER THE FIRING OF RULE ESTABLISH_SITE_TOPOGRAPHY_TYPE

```
Working Memory Configuration Prior to Firing of Rule

TOPOGRAPHY_OF_THE_SITE
     ^landform_type                      sandstone_humid
     ^topography                         steep_slopes
     ^certainty_value_of_topography      +2
     ^status                             nil

TOPOGRAPHY_OF_THE_SITE
     ^landform_type                      sandstone_humid
     ^topography                         medium_slopes
     ^certainty_value_of_topography      +1
     ^status                             nil

LANDFORM_TOPOGRPHY_PAIR
     ^landform_type                      sandstone_humid
     ^topography                         steep_slopes
     ^landform_topography_peh            0.60
     ^landform_topography_penoth         0.0
     ^status                             nil

LANDFORM_TOPOGRPHY_PAIR
     ^landform_type                      sandstone_humid
     ^topography                         medium_slopes
     ^landform_topography_peh            0.20
     ^landform_topography_penoth         0.0
     ^status                             nil
```

```
Working Memory Configuration After the Firing of Rule

TOPOGRAPHY_OF_THE_SITE
     ^landform_type                      sandstone_humid
     ^topography                         steep_slopes
     ^certainty_value_of_topography      +2
     ^status                             done

LANDFORM_TOPOGRAPHY_PAIR
     ^landform_type                      sandstone_humid
     ^topography                         steep_slopes
     ^landform_topography_peh            0.60
     ^landform_topography_penoth         0.0
     ^status                             final
```

The way chosen for testing TAX for the consistency and accuracy of the embedded knowledge with that of experts, and for the accuracy and correctness of its conclusions was to ask potential analysts to assess the same site, with the same sources of data, using TAX, and then compare the different evaluations.

Appendix I shows a listing of a terminal session with the prototype Terrain Analysis Expert (TAX) system. After the analyst logged into the system, he loaded the OPS5 language interpreter, and then he executed the TAX program (ETA6). The example shown concerns the identification of a humid sandstone. Input data were provided by Olin W. Mintzer [22]. Human problem solving for the same site has been described by Mintzer and Messmore [21]. The analyst's input consisted of the physiographic section of the site and his interpretation of the pattern element values and certainties. The analyst's input is underlined.

In all there are 45 production rules in TAX. The statistics at the end of the sample run (Appendix I) show that there were 138 firings, i.e., the recognize-act cycle was executed 138 times, some rules firing more than once. The average size of the working memory for this particular run was 114. The average size of the conflict set was 10. This number gives an approximate idea of the external control exercised over the firing of the rules by the structure and design of the production rules. For instance, a mean conflict set size of 1, implies that the program is fully procedure driven. In such a case it might be more efficient to code the program in a traditional programming language like LISP, C etc.

In the present implementation the control exercised by TAX's rules is minimal. It mainly assists in 1) directing the reasoning in backward form, and 2) assuring that context-sensitive rules would fire in the appropriate logical order. For example, firstly, the pattern element values are queried from the analyst, secondly the best values are selected, thirdly the $P(E/^{-}H)$ values are computed, and finally the a priori certainties of the hypothesized landforms are updated to their a posteriori values.

CONCLUSIONS

The implementation of this research prototype has shown that the representation of knowledge in a production system model is appropriate for the task domain. By utilizing knowledge engineering techniques, a prototype system was built that performed well on some tasks of the domain.

A number of features of the system are notable. The development of the system led to a formal description of the terrain analysis problem solving process. TAX is the first expert system prototype that emulates the terrain analysis process for identification of landforms.

One gains flexibility but loses readability by using a language instead of an expert system tool. For example, in the current version of OPS5 language, rules cannot be written in natural language, which could have made the program readable and understandable by any analyst. Also, OPS5 does not provide for extensive explanation facilities. On the other hand, OPS5 provides the freedom in designing control strategies on top of the recognize-act-cycle or other desirable forms of reasoning and uncertainty handling mechanisms.

The architecture of the system is modular and this makes it flexible and easily extendable. In contrast to a procedural computation in which terrain analysis knowledge would have been mixed in with instructions about the flow of control, an expert system approach allowed separation of the knowledge (in the knowledge base) from the control (provided by the inference engine).

Rules in a production system program, like TAX, are applied following a forward-chaining strategy, that is, the conditions in a rule specify the combinations of facts or objects to be matched against the current situation in the working memory. Consequently, TAX is a data-driven program instead of an instruction-driven program written in most procedural languages; that is, the communication among the rules is only by way of data and facts. Hence, unlike an algorithmic computer program, TAX can be easily updated. In fact, there is no need for rewriting the program to include a new landform type. This feature makes TAX an ideal environment for incremental programming of terrain-related knowledge, which is essential in the early stages of modeling such a complex process.

In many instances, the initial programming of a process based on an expert system paradigm has illustrated a well-structured line of reasoning, which in turn, has led the model developers to more efficient algorithmic programs. Upon extensive experimentation of future versions of the TAX system, it might be discovered that such an approach is desirable. In that event a procedural representation of the problem may be selected. However, since object, rule, and frame-based representations appear to provide for a more direct representation and acquisition of the expert's knowledge, it is highly improbable that a procedural approach will be preferable, at least, in the stage of system development.

The premise, in choosing OPS5, was that a production system architecture could represent terrain analysis problem solving, and that the experience gained through such an experiment could be used as a vehicle for discovering what new methodology needs to be employed. The utility of other problem-solving paradigms and expert system tools for terrain analysis are under investigation. This first formulation of the terrain analysis problem in OPS5 has provided a tangible point of reference to compare with human performance and other knowledge representation schemes to be implemented in the future.

The work reported here was supported partially by two Louisiana Department of Transportation Research Center projects (Contracts 87-2GT and 87-1H(B)) and by the Department of Civil Engineering and the Remote Sensing and Image Processing Laboratory at Louisiana State University. The assistance of Dr. Roger K. Seals, Professor Ara Arman, and Dr. Charles A. Harlow is greatfully acknowledged. For his help and expertise we thank Olin W. Mintzer, Research Civil Engineer at the Engineering Topographic Laboratories. He has kindly offered his estimates for both the probabilities of the knowledge base concerning the landform-pattern element pairs and his estimation of certainty values of the pattern element values of the test site. The assistance of R. Greene and R. Czek in interpreting the test site and providing their certainty values in testing the system is greatly appreciated. We thank Susan Sartwell for the excellent typing of the final text, tables, and figures.

REFERENCES

1. Adeli, H., (ed.), *Expert Systems in Construction and Structural Engineering*. Chapman and Hall, London, U.K., 1988.
2. Argialas, D., Computer-assisted recognition of drainage patterns. Technical Paper, Annual Convention of American Society for Photogrammetry and Remote Sensing, Vol. 4, pp. 435–444, 1986.
3. Argialas, D., Lyon, J., and Mintzer, O., Quantitative description and classification of drainage patterns. Photogrammetric Engineering, and Remote Sensing. Vol. 54, No. 4, April 1988.
4. Argialas, D. and Narasimhan, R., TAX: A prototype expert system for terrain analysis. Journal of Aerospace Engineering, ASCE, Vol. 1, No. 3, July 1988.
5. Brownston, L., Farrell, R., Martin, N., and Kant, E., *Programming Expert Systems in OPS5*, Addison-Wesley, Reading, MA, 1985.
6. Buchanan, B. and Shortliffe, E., (eds.), *Rule-Based Expert Systems: The MYCIN Experiments of the Stanford Heuristic Programming Project*. Addison-Wesley, Reading, MA, 1984.
7. Charniak, E. and McDermott, D., *Introduction to Artificial Intelligence*. Addison-Wesley, Reading, MA, 1985.
8. Chestek, R., Muller, H., and Chelberg, D., Knowledge-based terrain analysis. *Applications of Artificial Intelligence II, SPIE*, Vol. 548, pp. 46–56, 1985.
9. Duda, R., The PROSPECTOR system for mineral exploration. Final Report, SRI International, Menlo Park, CA, 1980.
10. Forgy, C., OPS5 user's manual. Technical Report CMU-CS-81-135, Carnegie-Mellon University, Pittsburg, PA, 1981.
11. Hall, C. and Benz, S., A knowledge-based photointerpretation system. *National Defense* 69(404):54–62 (1985).
12. Harmon, P. and King, D., *Expert Systems: Artificial Intelligence in Business*. Wiley & Sons, New York, NY, 1985.
13. Hayes-Roth, F., Waterman, D., and Lenat, D., *Building Expert Systems*. Addison-Wesley, Reading, MA, 1983.
14. Jackson, P., *Introduction to Expert Systems*, Addison-Wesley, Reading, MA, 1986.
15. Kostem, C., and Maher, M., (eds.), *Expert Systems in Civil Engineering*. American Society of Civil Engineers, New York, 1986.
16. Leighty, R., A logical approach toward terrain pattern recognition for engineering purposes. Ph.D. Dissertation, The Ohio State University, Columbus, OH, 1973.
17. Leighty, R., Research for information extraction from aerial imagery, U.S. Army Corps of Engineers, Remote Sensing Symposium, Engineer Topographic Laboratory, Reston, VA, 1979.
18. Lillesand, T. and Kiefer, R., *Remote Sensing and Image Interpretation*. John Wiley & Sons, 1979.
19. Maher, M., (ed.), *Expert Systems for Civil Engineers*. American Society of Civil Engineers, New York, 1987.
20. McKeown, D., Knowledge-based aerial photointerpretation. *Photogrammetria* 39:91–93 (1984).
21. Mintzer, O. and Messmore, J., Terrain analysis procedural guide for surface configuration. U.S. Army Corps of Engineers, Engineering Topographic Laboratories, Fort Belvoir, VA, 1984.
22. Mintzer, O., Personal communication, 1987.
23. Reboh, R., Knowledge engineering techniques and tools in the PROSPECTOR environment. Technical Note 243, SRI International, Menlo Park, CA, 1981.
24. Reboh, R., Reiter, J., and Gaschnig, J., Development of a knowledge-based interface to a hydrological simulation program. SRI International, 1982.
25. Shortliffe, E., *Computer-Based Medical Consultations: MYCIN*. Elsevier, New York, NY, 1976.
26. Way, D., *Terrain Analysis*. Hutchinson and Ross, 1978.
27. Winston, P., *Artificial Intelligence*. Addison-Wesley, 1984.

APPENDIX: A TYPICAL CONSULTATION SCRIPT GENERATED WITH THE TERRAIN ANALYSIS EXPERT (TAX) SYSTEM.

```
Script started on Wed Nov 11 12:41:23 1987

% ops5
→ (load 'eta6)

[load eta6]

**********************************************t

→ (strategy mea)
mea

→ (make start)

nil

→ (run)

1. start 1

Please provide the following information about the site.

To which Physiographic-section does the site belong?

cumberland-plateau

2. create-knowledge-base 4

3. hypothesis-landform-section-make 8

4. site-gully-amt 75 73

Is the gully-amt of the site

none ? Give a certainty value between -3 to 3

-3

5. site-gully-amt 78 72

Is the gully-amt of the site

few ? Give a certainty value between -3 to 3

1
```

```
6. site-gully-type 83 71
Is the gully-type of the site
v-shaped ? Give a certainty value between -3 to 3
3
7. site-landuse-valleys 88 69
Is the landuse-valleys of the site
cultivated ? Give a certainty value between -3 to 3
-1
8. site-landuse-valleys 93 68
Is the landuse-valleys of the site
forested ? Give a certainty value between -3 to 3
3
9. site-landuse-slopes 98 66
Is the landuse-slopes of the site
cultivated ? Give a certainty value between -3 to 3
-3
10. site-landuse-slopes 103 65
Is the landuse-slopes of the site
forested ? Give a certainty value between -3 to 3
3
11. site-soil-tone 108 63
Is the soil-tone of the site
medium ? Give a certainty value between -3 to 3
1
12. site-soil-tone 113 62
Is the soil-tone of the site
light ? Give a certainty value between -3 to 3
0
```

```
13. site-soil-tone 118 61
Is the soil-tone of the site
dark ? Give a certainty value between -3 to 3
0
14. site-drainage-txtr 123 60
Is the drainage-txtr of the site
coarse ? Give a certainty value between -3 to 3
3
15. site-drainage-type 128 59
Is the drainage-type of the site
internal ? Give a certainty value between -3 to 3
-2
16. site-drainage-type 133 58
Is the drainage-type of the site
angular ? Give a certainty value between -3 to 3
2
17. site-topography 138 56
Is the topography of the site
steep-slopes ? Give a certainty value between -3 to 3
3
18. site-topography-establish 141 145
19. site-topography-penoth-compute 147 149 31 7
20. site-topography-penoth-compute 151 149 11 6
21. site-topography-hypothesis-update 143 149 154
22. site-drainage-type-establish 136 140
    ...................................
    ...................................
    ...................................
```

```
60. site-soil-tone-infer 301 40 116


61. site-gully-amt 306 53

Is the gully-amt of the site

many ? Give a certainty value between -3 to 3

-2

62. site-topography-establish 294 298

63. site-topography-penoth-compute 315 317 154 8

    ..................................

    ..................................

    ..................................


137. site-gully-amt-hypothesis-update 577 583 588

138. display-landform 2 595

The site appears to be ****** sandstone-humid ******

The certainty associated with this result is

*****  0.99   *****

end -- no production true

 45 productions (415 // 711 nodes)

 138 firings (594 rhs actions)

 114 mean working memory size (156 maximum)

 10 mean conflict set size (33 maximum)

 275 mean token memory size (377 maximum)

nil

→ ^D

Goodbye

%

script done on Wed Nov  11 12:45:11 1987
```