Partial Automation of the Cartographic Design Process

Richard Smith*

* Texas A&M University – Corpus Christi

Abstract. Individuals once unable to participate in the map making process find themselves with the technology, if not the cartographic knowledge, to map whatever phenomena they desire. With this democratization of mapping comes a larger community of map makers and increased production and consumption of poorly designed maps. An enlarged map making community, coupled with an increase in the consumption of maps, amplifies the possibility for the consumption of poorly designed maps and the public perception of what a well-designed map is, thus, creating a positive feedback loop reducing the quality of maps that will be created in the future. To introduce negative feedback, map makers must be educated in cartography and cartographic design. A software system has recently been created that automatically chooses the best symbols for a general reference map. This system relies on an expert system and data mining, to analyze and store vast amounts of information about traditional cartographic knowledge and structure of commonly used geospatial information, both of which are utilized to create a map. Without user intervention, the software markedly increases the cartographic quality of the map.

Keywords: Cartography, Democratization, Expert System

1. Introduction

To reach the partial automation of the cartographic design process, the symbol design decisions that cartographers make when constructing a map must be emulated by the computer. The challenge in having the computer emulate a cartographer is translating the human process of symbol design into rules and heuristics that can be acted upon by the computer. Cartographic design is an ill-structured problem that cannot be solved effectively by conventional computer programming. As conventional computer programming is not suited for emulating a cartographer's decisions, this research employs expert systems technology to store, evaluate, and report facts, rules, and results with regard to map symbol design. The purpose of expert systems is to assist users in a decision on the basis of stored knowledge and is designed to deal with ill-structured problems such as symbol design choices. This paper discusses an expert system that was created in order to emulate the decisions a cartographer makes when designing a general reference map.

2. Applicability of Expert Systems to the Cartographic Symbol Selection

2.1. Benefits of Expert Systems

For this research, expert systems are employed to assist new and experience map makers by automatically choosing cartographic symbols for data sets added to a map. Applying expert systems technology for choosing cartographic symbols offers three major benefits to users. The *first benefit* is that an expert system provides a user with the expert knowledge they seek without needing a human expert in the room. The *second benefit* is an expert system always being available whereas the human counterpart may be absent. The *third benefit* is that one expert system can contain the knowledge of multiple human experts, thus, acting as a human expert multiplier. In the context of map making, an expert system contains a knowledgebase of cartographic domain knowledge that it can intelligently apply user's unique mapping problem.

The ill-structured and heuristic nature of cartographic knowledge is an excellent domain for modeling in an expert system. However, an important caveat should be noted. It is easier to program expert systems with shallow knowledge based on experience and empirical knowledge than the complex underlying causes. A heuristic is an experience-based technique for problem solving when an exhaustive search for a solution is impractical (Russell and Norvig 2003). Heuristic methods allow the expert system to remain shallow in knowledge as a heuristic does not require perfect data or perfect solutions; it aids in the solution with a guarantee of success. As heuristics provide a "rule-of-thumb" method, and can be based on generally correct assumptions and do not require expensive overhead in knowledge modeling, a heuristic approach has been adopted for the facts and rules in the expert system created for this research.

2.2. Obtaining Expert Knowledge

As the cartographic expert system is being designed to emulate the expert knowledge of human cartographers, a translation from human knowledge to computer structures must be undertaken. For this research, three sources of human knowledge of cartography were targeted: cartographers, cartography textbooks, and published general reference maps. To remove the potential for bias, each source was evaluated against the other sources of knowledge to identify commonalities so that the "safest" cartographic knowledge was obtained.

With the sources of cartographic knowledge identified, the iterative process of obtaining the knowledge began. The cartographic expertise was initially recorded informally in the form of IF-THEN statements and lists of commonly used visual variables.

The IF-THEN statements and lists were next converted into facts and rules structures useable by the expert system. Typically, for each cartographic concept obtained from expert sources, both a fact, and a rule would be created; the fact representing the cartographer's knowledge, and the rules representing actions the cartographer would take based on their knowledge.

2.3. Building Facts and Rules

To illustrate how this process of translating knowledge to expert system data structures, an example will be presented. This example discusses how the problem of dataset draw ordering was represented and solved in both the informal and expert system structure.

In this example, published general reference maps were evaluated to determine which datasets were placed in what draw order. For each map, the draw order was recorded in a list format similar to the list shown in *Figure 1*. After the maps were analyzed, cartography textbooks were referenced in regard to layer draw orders. Three cartography textbooks were referenced for this research: "Cartography: Thematic Map Design" (Dent, Torguson, Hodler 2009), "Elements of Cartography" (Robinson et al. 1995), and "Thematic Cartography and Geovisualization" (Slocum et al. 2008). In these textbooks, figures of maps were analyzed, and chapters on design of reference maps, layout design, and figure ground relationship were referenced for relevant information. Lastly, the compiled list of information was evaluated by cartographers to determine, in general, which theme would be placed at what position in the drawing order of the map. The result from the cartographers was a single IF-THEN statement and an ordered list representing the determined draw order of each dataset theme as displayed in *Figure 1*.

General Drawing Order of Dataset Themes

1. Hospitals
2. Emergency
3. Airports
4. Roads
5. Railroads
6. Trails
7. Contours
8. Water
9. Parks
10. StateCounty

Ordering Logic

IF Target Dataset Theme has a higher draw order

(lower number) than a Dataset Theme above

THEN place Target Dataset Theme one place above.

Figure 1. Obtained Cartographic Information in Initial IF-THEN and List Format

With the cartographic knowledge paired down to the final version informal version, it was converted into the facts and rules structures used by the expert system as displayed in *Figure 2*. The expert system structure required three parts for this translation: a template for the knowledge (deftemplate), the knowledge (deffacts), and the rule (defrule). The deftemplate and deffacts were converted from the list of the drawing orders displayed in *Figure 1*, and the deffacts were converted from the IF-THEN statement.

```
(deftemplate MAIN::theme-ordering
   (slot theme (default ?NONE))
   (multislot above))

(deffacts MAIN::initial-theme-ordering
   (theme-ordering (theme Parks) (above StateCounty))
   (theme-ordering (theme Roads) (above Water Parks StateCounty
        Trails))
   (theme-ordering (theme Water) (above Boundaries Parks))
   (theme-ordering (theme StateCounty) (above))
   (theme-ordering (theme Contours) (above Parks StateCounty))
   (theme-ordering (theme Airports) (above Parks Roads Water
        StateCounty Contours Trails Railroads))
   (theme-ordering (theme Trails) (above Contours Parks StateCounty))
   (theme-ordering (theme emergency) (above Parks Roads Water
        StateCounty Contours Trails Railroads))
   (theme-ordering (theme Hospitals) (above Parks Roads Water
        StateCounty Contours Trails Railroads))
   (theme-ordering (theme Railroads) (above Parks Roads Water
        StateCounty Contours Trails)))

(defrule MAIN::order-mapLayers
   ?mapLayer1 <- (mapLayer (theme ?theme) (drawOrder ?loc1))
   (theme-ordering (theme ?theme) (above $? ?abovetheme $?))
   ?mapLayer2 <- (mapLayer (theme ?abovetheme) (drawOrder ?loc2))
   (test (> ?loc1 ?loc2))
   =>
   (modify ?mapLayer1 (drawOrder ?loc2))
   (modify ?mapLayer2 (drawOrder ?loc1)))
```

Figure 2. Obtained Cartographic Information in Expert System Structure

This process of obtaining cartographic knowledge was repeated for representing the concept of a map layer, describing colors, choosing colors for each theme, assigning symbology for each theme, and defining and setting the map scale. In all cases, these concepts were recorded in an informal fashion to maintain a high level of human readability. Only after all sources of knowledge have been referenced would the process of translating the knowledge into an expert system structure take place.

3. Methodology

Expert systems are “an intelligent computer program that uses knowledge and inference procedures to solve problems that are difficult enough to require significant human expertise for their solution” (Edward Feigenbaum in Giarratano and Riley 2005, 5). That is, the goal of an expert system is to emulate the thought process of a human to arrive to the same solution of a human expert. At a high level, expert systems are composed of two parts: a knowledge base, and an inference engine. The expert system uses the knowledge stored in the knowledge base, and facts entered by the user to infer an expert answer using the inference engine. Expert systems have been used for many different applications in many different knowledge areas and have successfully emulated the decisions made by experts.

This methodology section is split in two sub-sections. The first sub-section, CLIPS Expert System, will describe relevant aspects of the expert system software used to build the cartographic expert system. The second sub-section, Program Design of Cartographic Expert System, will detail the expert cartographic knowledge base and rules programmed into the expert system.

3.1. CLIPS Expert System

The expert system C Language Integrated Production System (CLIPS) was chosen for this research. “CLIPS is a multi-paradigm programming language that provides support for rule-based, object-oriented, and procedural programming.” (Giarratano and Riley 2005) CLIPS was originally designed as a rules-based production system, but later included procedural and object-oriented language support.

CLIPS is composed of many components, however, seven components are primarily used in this research as they provide all the required functionality. These seven components are: facts, fact templates, fact list, default facts, knowledge base, agenda, and inference engine. Each of these seven components will now be discussed briefly so that the reader can understand the translated cartographic knowledge presented later.

A **fact** is an item of knowledge and may be entered by a user, or stored as a default item of knowledge used to initialize the expert system. In CLIPS, facts are represented by a relation name, followed by zero or more slots. A relation name is a symbol (contiguous string of printable characters) that relates to a fact template (fact templates discussed below). Slots are sets of key/value pairs of information and each slot may contain multiple values related to a single key. *Figure 3* displays an example of a fact stored in CLIPS.

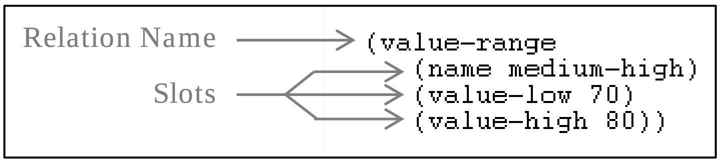

Figure 3. Example Fact Stored in CLIPS

A **fact template** is a defined data structure that specifies a list of valid slots for a given relation name. Facts must have a fact template defined before they are allowed to be asserted (inserted) into the fact list. In CLIPS, a fact template is known as a “deftemplate” which is short for “default template”. Deftemplates are static data structures that are not typically created or deleted during or between runs of an expert system program. Each deftemplate must have a relation name specified. Any fact that wishes to use the data structure defined by a deftemplate must start with the same relation name as the deftemplate. The basic syntax for a deftemplate is shown in *Figure 4*.

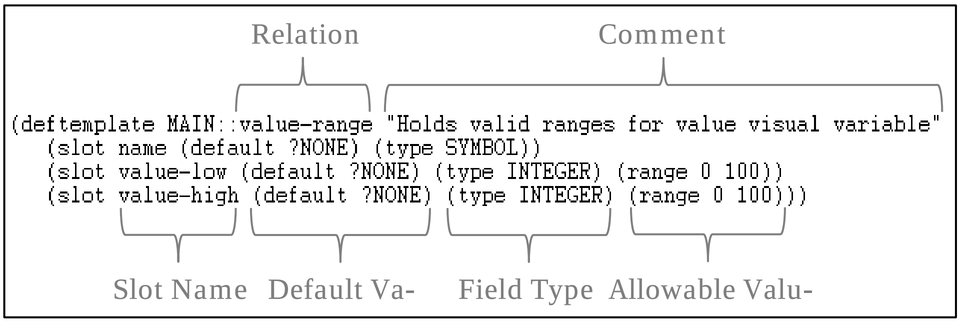

Figure 4. Basic Syntax of deftemplate

Fact lists contains a record of all information with which the expert system can reason. In a fact list, each fact is provided a fact address which provides a way for the expert system to uniquely identify each fact. The fact list acts as the central repository for all knowledge that the expert system will reason with. Facts in the fact list can be inserted, deleted, or modified by users or expert system rules.

Default facts (“deffacts” in CLIPS) are facts that exist upon initialization of the expert system. Default facts seed the expert system with knowledge about the world state. This default world state could be the initial state of a problem set which the expert system is meant to solve, and/or facts relating to rules or experience contributed by knowledge domain experts.

The **knowledge base (rules)** of an expert system contains rules that execute when facts are entered into the fact list that meet the rule's execution requirements. Each rule is provided a name to differential the rules to CLIPS.

Rules are entered into CLIPS as a "defrules" construct and is composed of two parts separated by an arrow symbol (=>): the left-hand side (LHS) which specifies conditional elements, and the right-hand side (RHS) which lists actions. The conditional elements listed in the LHS are a list of pattern matching constraints meant to match values in fact slots. The LHS can have one or more conditional elements. Only when all conditional elements on the LHS match facts in the fact list will the rule be activated and placed on the agenda.

As an example of a rule, the "find-Joe" rule is listed in *Figure 5* and has one conditional element in the LHS and one action in the RHS. The conditional element in the LHS matches facts related to the "Person" deftemplate. Specifically, the conditional element matches against one slot in the "Person" facts: the first-name slot. In this case, the LHS will match a "Person" fact that contains the symbol "Joe" in the first-name slot. With the LHS finding a matching fact, the rule activates and is placed on the agenda. When the agenda is run, one statement is printed stating: "The person 'Joe' was found."

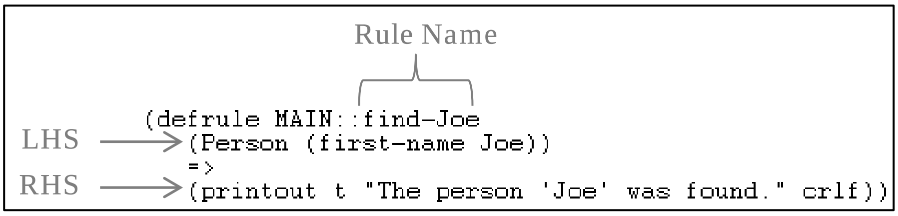

Figure 5. find-Joe Rule

In CLIPS, the **agenda** is the collection of all activated rules. A rule is considered activated when all conditional elements match patterns found in facts in the fact list. The agenda acts as an execution queue for when the expert system is run. When the expert system is run, the activated rules on the agenda are fired and the RHS of each activated rule is executed. When multiple rules exist on the agenda (which is often the case), the inference engine (discussed below) determines which rule to fire.

The **inference engine** determines when a rule on the agenda fires. The inference engine matches the facts in the fact list against the conditional elements on the LHS of each rule and arranges the activated rules on the agenda based on its salience value.

3.2. Program Design of Cartographic Expert System

This section details a few examples of the deftemplates, deffacts, and defrules programmed into the CLIPS expert system for the purpose selecting symbology for geographic data sets. The input to the expert system is basic information about a data set such as: theme, geometry, data set name, and map scale. To make the input as simple as possible to the user, the expert system receives three of these inputs from the program designed in Smith (2011) leaving scale the only user input. From this basic input information, the expert system will determine draw order, and symbol design for the data sets and return the results of its design decisions.

3.2.1. Example deftemplates in the Cartographic Expert System

Eleven deftemplates were created for the cartographic expert system, each providing a template for holding important cartographic knowledge from expert sources and information about the input geospatial data sets. Two deftemplates will be presented here to provide an idea of how cartographic knowledge was translated into the CLIPS expert system.

The *mapLayer* deftemplate holds information about the input data set that will be rendered as a layer in the map. The *mapLayer* deftemplate is displayed in *Figure 6*. There are fourteen slots in the *mapLayer* deftemplate. The first three slots, name, theme, and geometry are the only required slots that must be populated by the user and serve as the only input into the expert system. The remaining eleven slots hold the draw order and symbolization choices.

```
(deftemplate MAIN::mapLayer
      (slot name (default ?NONE))
      (slot theme (default ?NONE) (allowed-symbols Airports
            emergency Contours Roads Trails Railroads State-
            County Water Parks Hospitals))
      (slot geometry (default ?NONE)
            (allowed-symbols point line polygon))
      (slot drawOrder (default 0))
      (slot hue-value (default none))
      (slot saturation-value (default none))
      (slot value-value (default none))
      (slot symbol-size)
      (slot symbol (default none))
      (slot line-width)
      (slot line-type)
      (slot outline-hue-value)
      (slot outline-saturation-value)
      (slot outline-value-value))
```

Figure 6. mapLayer deftemplate

The *theme-hsv-color-preferences* deftemplate holds the preferred color information for each map theme and is shown in *Figure 7*. The purpose of this deftemplate is to hold the preferred color information for each theme's symbols. There are five slots in this deftemplate: geometry, theme, hue, saturation, and value. Geometry is designated in the case that a cartographer would have a different color preference based on geometry type.

```
(deftemplate MAIN::theme-hsv-color-preferences
      (slot theme (default ?NONE) (allowed-symbols
            Airports emergency Contours Roads Trails
            Railroads StateCounty Water Parks Hospitals))
      (slot geometry (default ?NONE)
            (allowed-symbols point line polygon))
      (multislot hue (default ?NONE))
      (multislot saturation (default ?NONE))
```

Figure 7. theme-hsv-color-preferences deftemplate

3.2.2. Example deffacts in the Cartographic Expert System

The role of a deffact in CLIPS is to seed the expert system with knowledge. For the cartographic expert system, the deffacts represent expert cartographic knowledge that will be applied to map layers input by the user. The deffacts in the cartographic expert system comprise the largest portion of the expert system. The deftemplates were designed to allow for easy growth in cartographic knowledge by minimizing the amount of hard-coded information in the deftemplates; this allows the expert system to easily include additional deffacts without having to restructure or redesign deftemplates or the expert system. This section will discuss a representative sample of the deffacts included in the expert system.

The *initial-theme-ordering* collection of deffacts (*Figure 8*) stores in map layer ordering information based on its theme. Every theme listed in the slot in a fact represents every theme that should be placed underneath the current map layer.

```
(deffacts MAIN::initial-theme-ordering
      (theme-ordering (theme Parks) (above StateCounty))
      (theme-ordering (theme Roads)
            (above Water Parks StateCounty Trails))
      (theme-ordering (theme Water) (above Boundaries Parks))
      (theme-ordering (theme StateCounty) (above))
```

Figure 8. Sample of initial-ordering deffacts

Initial-color-preferences collection of deffacts stores the color preferences for a given theme of data. These facts represent preferred colors for each theme based on basic cartographic color theory. For instance, themes that cover large areas, such as the StateCounty theme are given a cooler, low saturation color to create a nice ground for the map while figures, such as airports, and water are provided with warmer, medium to high saturation colors to create nice figures and a good contrast from the ground. A sample of the theme color preferences is listed in *Figure 9*.

```
(deffacts MAIN::initial-color-preferences
      (theme-hsv-color-preferences (theme Water)
            (geometry polygon) (hue blue) (saturation high)
            (value medium-high))
      (theme-hsv-color-preferences (theme Trails)
            (geometry line) (hue brown) (saturation high)
            (value high))
      (theme-hsv-color-preferences (theme Parks)
            (geometry polygon) (hue green) (saturation high)
            (value medium-high))
```

Figure 9. Sample of initial-color-preferences deffacts

3.2.3. Example defrules in the Cartographic Expert System

The role of a defrule in CLIPS is to execute some action when facts in the fact list exist that meet the conditional elements of the rule. For the cartographic expert system, seven defrules represent the actions that are to be taken by the expert system in creating the basic general reference map. This section will discuss two defrules and how they work. For each defrule code listing, line numbers have been added to assist in referencing specific lines in the discussion.

The *order-mapLayers* defrule (*Figure 10*) arranges the map layers' draw order based on the theme ordering preferences stored in facts that relate to the *theme-ordering* deftemplate. The order-mapLayers defrule has a four-part conditional element shown in lines 2 thru 5 in *Figure 10*. Line 2 matches a map layer storing its theme and drawOrder in the ?theme and ?loc1 variables respectively. Line 3 looks for a fact that specifies which themes should be draw below the current theme. It performs this match by matching the value stored in ?theme against facts that reference the theme-ordering deftemplate. Once a match has been found for theme ordering, one of the themes listed as being below the current theme is stored in the ?abovetheme variable. Line 4 uses the value stored in ?abovetheme to identify any map layer that has the theme that should be draw below the map layer identified in line 2. If a map layer is found, its location is stored in ?loc2. Line 5 tests to see if the draw order of the first map layer is greater than the draw order of the second map layer. If the second map layer's draw order is lower, then the rule activates and lines 7 and 8 switch the draw order numbers between the two map layers.

```
1(defrule MAIN::order-mapLayers
2     ?mapLayer1 <- (mapLayer (theme ?theme) (drawOrder ?loc1))
3     (theme-ordering (theme ?theme) (above $? ?abovetheme $?))
4     ?mapLayer2 <- (mapLayer (theme ?abovetheme) (drawOrder ?loc2))
5     (test (> ?loc1 ?loc2))
6     =>
7     (modify ?mapLayer1 (drawOrder ?loc2))
8     (modify ?mapLayer2 (drawOrder ?loc1)))
```

Figure 10. order-mapLayers defrule

The *match-theme-colors* defrule shown in *Figure 11* sets all map layers of the same theme to the same hue, saturation, and value so the theme has a unified color on the map. This defrule has a two-part conditional element. The first part of the conditional element on line 2 identifies a map layer that has a hue, value, and saturation assigned. The second part of the conditional element shown on line 3 finds a map layer that has the same theme and geometry as the first map layer and does not already have the same hue, saturation, and value as the first map layer. With the conditional elements matched, the rule will activate and set the hue, saturation, and value of the second map layer equal to the hue, saturation, and value of the first map layer as shown on line 5.

```
(defrule MAIN::match-theme-colors
     (mapLayer (theme ?theme) (geometry ?geometry)
           (hue-value ?hue-value&~none)
           (saturation-value ?sat-value&~none)
           (value-value ?val-value&~none))
     ?layer2 <- (mapLayer (theme ?theme) (geometry ?geometry)
           (hue-value ~?hue-value)
           (saturation-value ~?sat-value)
           (value-value ~?val-value))
     =>
     (modify ?layer2 (hue-value ?hue-value)
           (saturation-value ?sat-value)
           (value-value ?val-value)))
```

Figure 11. match-theme-colors defrule

4. Results and Discussion

This section displays and discusses six map outputs from the cartographic expert system; three large scale, and three small scale maps to show that the expert system can symbolize differently at difference scales. The assumption is made that the map user has selected data appropriate for mapping at the chosen scale. In order to demonstrate the expert system's capacity to homogenize colors across map layers, some themes will be split across multiple map layers. In addition to the six maps generated by the cartographic expert system, two map outputs will be discussed that used the randomly assigned colors provided by the mapping software to provide a before and after view of the maps.

Before running the expert system, two maps were saved in order to visualize how the data was symbolized in the mapping software before the expert system was run. These maps are shown in *Figure 12(a)* and *Figure 13(a)*. These maps are examples of what is often presented to the user of the map software when adding data sets. The map layer orders are not optimal, as some layers obscure the view of underlying layers. The map maker will need to reorder and symbolize each layer in full to arrive at a pleasing map.

With the map layers added to the mapping software, the expert system was run three times at the same scale with the data sets in order to depict how each run of the expert system produces different results. The three map outputs are shown below in *Figure 12(b)(c)(d)*.

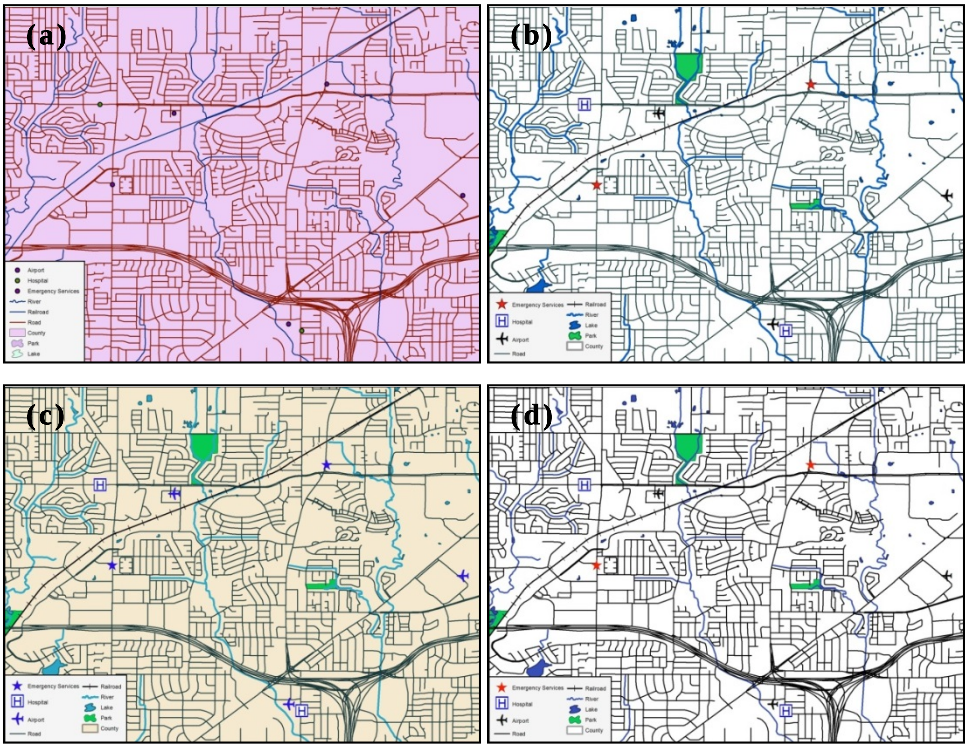

Figure 12. Default (a) and Expert System Designed Maps (b,c,d)

The second set of three maps was created at a map scale of 1:7,500,000 showing the entire state of Texas. As this is a large scale map, only five map layers were included to simulate what a reasonably-well populated general reference map would look like.

With the small scale map layers added to the mapping software, the expert system was run three times at the same scale and with the data sets show how each run of the expert system produces different results. The three map outputs are shown below in *Figure 12(b)(c)(d)*.

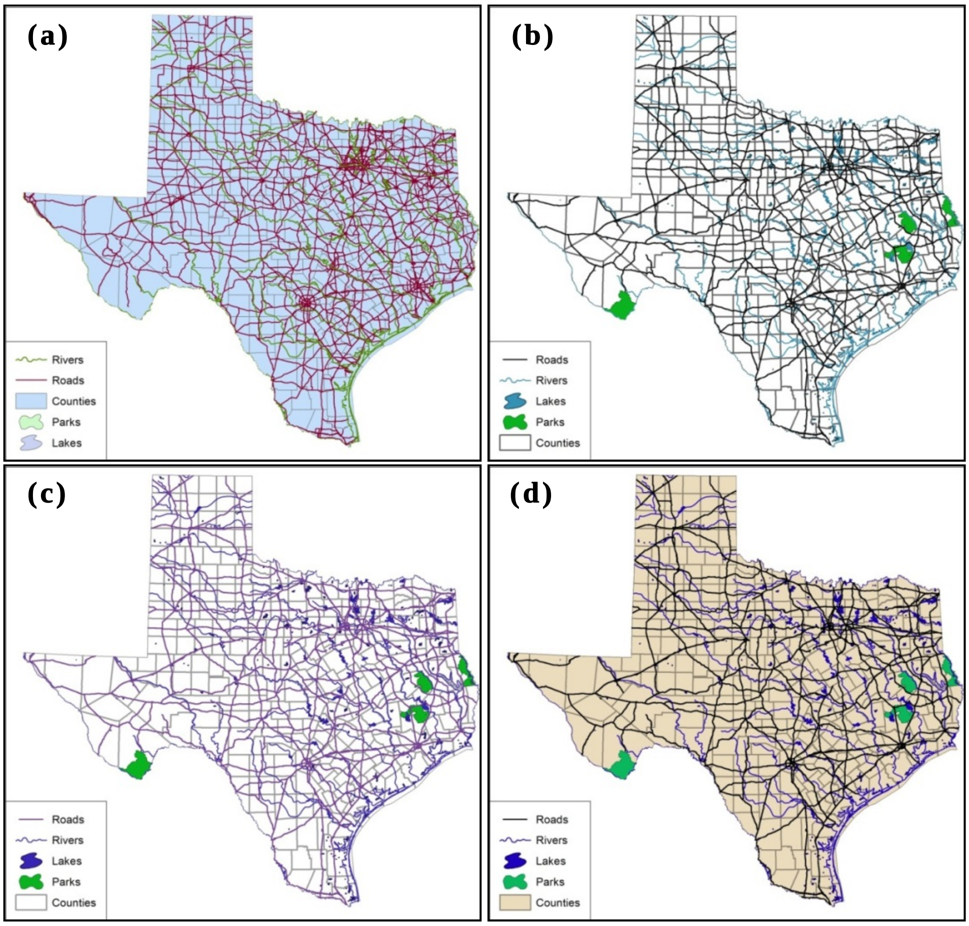

Figure 13. Default (a) and Expert System Designed Maps (b,c,d)

5. Conclusion

The results of this research show that there is promise partially automating the symbolization of a general reference map using a cartographic expert system. The six maps created by the expert system show that it can produce reasonably well-design general reference maps and would be a good starting point for the map maker opposed to the default, random colors chosen when the map layers are added to the map. While not all color combinations seemed to work completely, the resulting maps use safe enough colors and cartographic conventions to be used without any, or, minimal changes depending on the needs of the map maker.

While the results showed only the symbolization of a small and large scale map, the expert system is designed in a way to easily include other named scale ranges which would allow for insertion of additional expert cartographic knowledge. Multiple iterations of the expert system's facts, rules, and templates were undertaken to pare down its complexity to the expert system presented in this paper. Future expansion to the expert system should be reasonably straight-forward.

References

Dent, B. D., J. S. Torguson, and T. W. Hodler (2009) Cartography: Thematic Map Design Sixth ed. New York: McGraw-Hill.

Giarratano, J. C., and G. D. Riley (2005) Expert Systems, Principles and Programming. Course Technology.

Robinson, Arthur H., Joel L. Morrison, Phillip C. Muehrcke, A. Jon Kimerling, and Stephen C. Guptill (1995) Elements of Cartography 6th ed. New York: Wiley.

Russell, S., and P. Norvig (2003) Artificial Intelligence: A Modern Approach 2nd ed. Upper Saddle River, New Jersey: Prentice-Hall Inc.

Slocum, Terry A., Robert B. McMaster, Fritz C. Kessler, and Hugh H. Howard (2008) Thematic Cartography and Geovisualization 3rd ed. Upper Saddle River, New Jersey: Prentice Hall.

Smith, R. (2011) Data Mining Metadata to Automate the Identification of Datasets Commonly Found on a General Reference Map. International Cartographic Conference, July 4-8, 2011, Paris, France.