LEGAL EXPERT SYSTEM BUILDING: A SEMI-INTELLIGENT COMPUTER PROGRAM MAKES IT EASIER

by GRAYFRED B. GRAY,* BRUCE J. MACLENNAN,** JOHN E. NOLT,*** AND DONALD R. PLOCH****

I. INTRODUCTION

Consultant legal expert systems can enable lawyers to provide legal information, including simple reasoning, to other lawyers, clients, and the general public on an economical basis. A major obstacle to the development of legal expert systems, however, is the cost of the lawyer's time to build them. One way to make such expert systems more affordable is to eliminate the need for lawyers to become computer experts.[1]

This article demonstrates how a working prototype computer program,[2] Natural Language Expert System Builder (hereinafter NLESB), enables a lawyer to build a useful legal expert system in ordinary English without being a computer expert. First, the lawyer's use of NLESB

* Associate Professor of Law, University of Tennessee College of Law; B.A., 1961, Washington & Lee University; J.D., 1968, Vanderbilt University. Lead author.

** Associate Professor of Computer Science, University of Tennessee College of Liberal Arts; B.S., 1972, Florida State University; M.S., 1974, Purdue University; Ph.D., 1975, Purdue University.

*** Associate Professor of Philosophy, University of Tennessee College of Liberal Arts; B.A., 1972, Ohio State University; M.A., 1976, Ohio State University; Ph.D., 1978, Ohio State University.

**** Professor of Sociology, University of Tennessee College of Liberal Arts, B.A.; 1954, University of Virginia; B.D., 1957, Drew University; M.A., 1964, University of Connecticut; Ph.D., 1968, University of North Carolina, Chapel Hill.

[1] Another way would be to write laws, administrative rules, and other prescriptive legal documents in a form with which it is easier to build expert systems. *See* Cary G. deBessonet, *A Proposal for Developing the Structural Science of Codification*, 8 RUTGERS COMPUTER & TECH. L.J. 47 (1980-81), and Grayfred B. Gray, *Preparing Enacted Normalized Statutes For an Expert System*, 4 CC AI 389 (1987) (Belg.) [hereinafter *Enacted Normalized Statutes*]; Grayfred B. Gray, *Reducing Unintended Ambiguity in Statutes: An Introduction to Normalization of Statutory Drafting*, 54 TENN. L. REV. 433, 435 (1987) [hereinafter *Reducing Unintended Ambiguity*].

[2] The program is currently implemented in TurboProlog 2.0 and runs on an IBM-compatible PC with at least 640K RAM, DOS 3.0 or higher, and a hard disk.

as a semi-intelligent assistant is facilitated by a system of menus and the lawyer's use of ordinary English to write questions and laws. There is no computer language that the lawyer must master to build an expert system with NLESB as an assistant. A second important feature is that the lawyer builds the knowledge base with forms of expression, sentences, that are relatively easy to comprehend.[3] NLESB accepts rules in ordinary English, though in normalized form,[4] and parses[5] them into propositional data structures that it can use to draw inferences. NLESB has some features, particularly in its logic,[6] that are peculiar to the needs of legal expert systems.[7]

[3] Compare NLESB's sentences, as illustrated *infra* note 26 and accompanying text, to International Business Machine Company's Expert System Development Environment program, which requires the expert to use such structures as "THE—OBJECT—SHAPE." *See* EXPERT SYSTEM ENVIRONMENT: EXPERT SYSTEM DEVELOPMENT ENVIRONMENT USER'S GUIDE 3-4 (3d ed. 1988). Another expert program requires the following form:

```
If
     (date current) void ASK
     (contract date) void ASK
then
     (requirements formal satisfied):=
          ((date current) date-(contract date))
          smaller—than 00/03/00
```

A. Oskamp *et al*, *PROLEXS, DIVIDE and RULE:* a legal application, SECOND INTERNATIONAL CONFERENCE ON ARTIFICIAL INTELLIGENCE AND LAW 54, 61 (1989).

The authors are satisfied that such forms of expression of rules, while useful for their purposes, are obstacles to lawyer development of legal expert systems.

[4] *See infra* section II. A. for a discussion of normalized form.

[5] Parsing generally is the process of pulling apart the word and symbol units in a text and translating them into an underlying data structure. NLESB looks for five logical operators: NOT, AND, OR, IF . . . THEN, and IF AND ONLY IF . . . THEN. As part of the parsing process NLESB separates the word "not" from each proposition in which it appears so that the lawyer has to determine whether the proposition stated without "not" is the negation of the proposition stated with "not."

[6] NLESB's logic is not classical propositional logic, but a relevance logic that reflects legal reasoning better than classical propositional logic. There are significant differences from propositional logic with respect to conditionals and biconditionals and consequently differences with respect to other connectives. The authors are preparing an article describing this logic.

[7] While some expert system building tools impose severe limits on the lengths of rule names, NLESB gives a high degree of freedom in naming rules. NLESB can accept rule names up to 77 characters long. Thus the names can consist of legal citations, short names, derivations, or implications. The following are three kinds of names found in the orders of protection system whose development is illustrated in this article:

a. Def. of files pro se;
b. TCA sec. 36-3-601(1), -602 interp. 3 (1991) abuse in fact;
c. Impl.<TCA sec. 36-3-601(1), -602 (1991) threats.

Such flexibility in naming rules facilitates finding relevant rules during a search through the list of rule names and enhances the explanations NLESB provides for its actions. For example, when the lawyer testing the orders of protection system asked about issuing an *ex parte* order and NLESB reached a conclusion, NLESB explained its answer as follows:

NLESB performs the following general functions in building an expert system:

1. Takes in the rules,[8] aids the lawyer in insuring that they are in good form, and permits their review or modification at any time;
2. Helps the lawyer find propositions that mean the same thing, can be expressed better, or can link with other rules, and helps the lawyer create bridge rules[9] to link rules together or to improve user access to NLESB's knowledge;
3. Permits the lawyer to test the expert system interactively by posing hypothetical situations to the database and by posing questions as if the lawyer were a user of the system;
4. Enables the lawyer to annotate the rules as extensively as desired at the propositional level;
5. Provides explanations of the system's questions and conclusions as a by-product of the database itself without further work by the lawyer;
6. Generates an index that enables the lawyer (a) to locate all contexts in which a word appears in the database and (b) to direct the system to ignore certain common words in the statistical search routine described below, thereby avoiding spurious matches.

Using a statistical search routine,[10] NLESB engages the lawyer in-

---

> The court may immediately issue an ex parte order of protection [TCA SEC. 36-3-605(A(S1)) (3)] BECAUSE:
>
> > The petitioner files a petition for an order of protection [TCA SEC. 36-3-605(A(S1)) (1)], and
> >
> > The petitioner shows good cause for an order of protection [TCA SEC. 36-3-605(A(S1)) (2)].

By including the rule name, the explanation pinpointed exactly where in the rules each piece of information came from. The rule names are likely to be particularly helpful in maintaining the system.

[8] The lawyer can type rules and notes directly into NLESB or read them in from DOS text files created in a word processor. NLESB has a built in text editor for use in typing or revising rules and notes.

[9] We use the term "bridge rule" for rules that the lawyer adds to the system though they are not expressly stated in the law that is the basis of the expert system. The lawyer adds them to the rules either to facilitate entry to the system or to express implicit connections among the laws. Bridge rules always connect a rule either with knowledge or other rules. *See infra* note 37 and accompanying text for a discussion of an example of a bridge rule created to facilitate entry to the system.

[10] NLESB has used a statistical search routine since it was first developed in 1988. It is not the purpose of this article to justify the use of statistical matching processes for information retrieval in legal expert systems, but it bears noting that the utility of statistical search routines with natural language in legal materials has been recognized by West Publishing Company's "Westlaw Is Natural" (WIN) search technique in Westlaw. *See* Teresa Pritchard-Schoch, *WIN—WESTLAW Goes Natural*, 17 ONLINE 101 (1993); WEST PUBLISHING CO., NATURAL LANGUAGE SEARCHING 7 (1993). *See also* Howard R. Turtle and W. Bruce Croft, *A Comparison of Text Retrieval Models*, 35 COMPUTER J. 279 (1992) (suggesting that "the probabilistic approach is the current best theory for information retrieval"); Keller, *Good-Bye Teacher*, J. APPLIED BEHAVIOR ANALYSIS 79-89 (1968), *cited in*

teractively to identify logically related propositions that occur in different contexts and to state their logical relationships. The lawyer "unifies"[11] propositions, that is, identifies different propositions as logically equivalent, by searching and highlighting rules the lawyer wants to work on from menus listing all rules. NLESB uses its statistical match routine to decide whether particular propositions are candidates for unification. NLESB automatically disregards words that the lawyer has designated as not useful, including those that are so short that the lawyer decides they are unlikely to be useful in the match process.[12] Once it has found candidates, NLESB invites the lawyer to decide which propositions to consider unifying. If the lawyer chooses to consider a pair, NLESB offers the lawyer an opportunity to view all the contexts in which the candidates appear. Further, it offers an opportunity to choose one proposition to substitute for another, to keep them verbatim but store the fact that they mean the same thing, or to type in a new proposition to replace either of them.

This process improves NLESB's ability to draw inferences, while acutely focusing the lawyer's attention on possible inconsistencies, ambiguities, uncertainties, or other inadequacies in the way the law is expressed. It also helps the lawyer keep the expert system up to date and experiment with the effects of proposed changes in the law. Once the lawyer has written and polished the rules constituting the expert system's knowledge, the system with NLESB can stand alone as a consultant.

The consultant legal expert system is then available to provide information either to lawyer or non-lawyer "end users." In particular, NLESB with the expert system:

1. Answers direct questions by engaging in a dialogue with the end user;
2. Answers "what if" or hypothetical questions by engaging in a dialogue with the end user;

correspondence from Alan L. Tyree reporting use of a statistical matching process with SAGES (Short Answer General Examination System) as a watchdog in grading student short answer test questions with 80% correlation with the instructor's grading (Dec. 8, 1992 (copies available from the authors)). Artificial Intelligence Law-List (AIL-L) at AUSTIN.UNO.EDU (Dec. 9, 1992). There is discussion of WIN and Tyree's method on AIL-L (Sept.-Dec., 1992).

[11] We use "unification" in a sense that is different, though related, to the sense with which it is commonly used in artificial intelligence. There, the unification of two propositions refers to an automatic process of finding assignments to the variables of the propositions so that they become identical in form. However, since we use a propositional logic, which doesn't look within individual propositions, there are no variables. Therefore, in our system the unification of propositions results from the expert-system builder *declaring* that two propositions have identical meaning.

[12] The latter may be excluded as a class, e.g., all words less than four letters long.

3. Shows specific laws that lead to a given result in response to questions about "how" to get a particular result;
4. Conducts general searches for any combination of words in a proposition;
5. Provides the end user an index to learn the vocabulary of the knowledge base and other means to help locate relevant laws.

The same statistical search mechanism that helped the lawyer articulate logical relationships among the laws is used to direct natural language queries to the law, and NLESB's inference engine draws appropriate conclusions. Because there may be multiple propositions that match the end user's query, NLESB first inquires as to whether the information it has retrieved, or at least some part of it, is responsive to the request. Once the end user confirms that particular information is responsive, that is, the information is sufficiently similar that knowing more might answer the query, NLESB asks the user for further information needed to answer the query based on the rule that NLESB is applying. NLESB forms its questions directly from the propositions in the rules.[13]

At any point the end user can ask NLESB why it has asked a question, and NLESB explains by reference to the rule it is using to answer the query. Similarly, once NLESB has reached a conclusion, it can explain how it did so by reference to the facts provided by the user and the laws applied by NLESB.

In summary, to build a legal expert system from conventionally written laws with NLESB the lawyer performs three tasks:

1. Converts each law to a form with which NLESB can reason: A simple English, IF-THEN or IF-AND-ONLY-IF list of a set of factual conditions under which the law provides the stated results (a normalized law);
2. Improves the way the laws are expressed and connected to one another; and
3. Tests and revises the resulting expert system until, for its purposes, it is accurate, complete, and useable for its intended beneficiaries.

The first task is an intellectual one for the lawyer and may be efficiently done by the lawyer either at a word processor or by marking up

[13] The use of grammatically complete clauses for each condition and result is a characteristic of normalized rules, but is not necessary to NLESB's reasoning. *See Reducing Unintended Ambiguity*, *supra* note 1, at 436; Layman E. Allen & C. Rudy Engholm, *The Need for Clear Structure in 'Plain Language' Legal Drafting*, 13 U. MICH. J.L. REF. 455, 507 (1980) [hereinafter *Clear Structure*]. Such complete propositions make better questions to pose to the end user and minimize the programming that would otherwise be necessary to form questions from the rules. In the authors' experience building expert systems, complete propositions also make the rules clearer and facilitate deciding whether two differently worded propositions are identical in meaning.

text for typing into a word processor. It is important that the form in which the lawyer writes the rules be easily understood by the lawyer rather than being in a computer language. NLESB accepts rules in English sentences that non-lawyers can understand.[14] The first task is discussed in section II, below. The authors believe that the second and third tasks are better performed with a modestly intelligent computerized assistant such as NLESB. Those tasks are discussed in sections III and IV below.

This article illustrates a lawyer's use of NLESB by means of actual interactions that occurred between a lawyer and NLESB in building a consultant legal expert system. The examples resulted from building an expert system on the law of orders of protection, a procedure used in Tennessee in cases of domestic violence between adults in family-like settings.[15]

The article shows several ways in which NLESB helped the lawyer improve the intelligence, precision, clarity, and utility of the system. While NLESB has been developed to build consultant expert systems, experience building expert systems with it suggests that it can be helpful in writing statutes and legal documents that are intended to prescribe legal relationships.[16]

[14] It has been established empirically that non-lawyers can understand the sentences NLESB accepts as well as those in other forms and can understand NLESB sentences better when the sentences express complex rules. Donald R. Ploch et al., *Readability of the Law: Forms of Law for Building Legal Expert Systems*, 33 JURIMETRICS J. 189 (1993) (reporting an experiment performed in 1988) [hereinafter *Readability of the Law*]. *See also Clear Structure*, *supra* note 13, at 469-70 (reporting earlier experiments); *Enacted Normalized Statutes*, *supra* note 1, at 447. Peter Ziegler also argued for the need for such experimentation. *See* Peter Ziegler, *The Status of Normalized Drafting: The Need for Theory Building and Empirical Verification*, 27 OSGOOD HALL L.J. 337, 348 (1989).

The IF-THEN structure of English sentences for NLESB has also been shown to enable computer programmers to understand computer programs better. William J. Tracz, *Computer Programming and the Human Thought Process*, 9 SOFTWARE—PRACTICE AND EXPERIENCE 129 (1979).

[15] TENN. CODE ANN. § 36-3-601 et seq. (1991).

[16] See Reed Dickerson's discussion of computer aids for legal drafting in THE FUNDAMENTALS OF LEGAL DRAFTING § 12.1 (1986). Dickerson saw computer aids leading to improvement in the quality of drafting primarily from "closer attention to internal arrangement and completeness." *Id.* at 264. NLESB does indeed promote closer attention to both aspects of legal writing as is illustrated in sections II—IV, *infra*. *See also* Grayfred B. Gray, *An Experiment with Normalized Statutes in an Emycin Expert System*, *in* COMPUTING POWER AND LEGAL LANGUAGE 225 (Charles A. Walter ed., 1988), for an illustration of kinds of changes that were required to prepare normalized statutes for use in the EMYCIN expert system shell, and *Enacted Normalized Statutes*, *supra* note 1.

II. PUTTING LAW INTO THE LEGAL EXPERT SYSTEM

A. Converting Laws to Expressions with which NLESB Can Simulate Reasoning

The first step in developing the legal expert system was to state the law on protective orders against domestic violence between adults in a form with which NLESB could simulate reasoning. This involved converting conventionally written sentences into expressions of the conditions under which particular legal results follow.

Normalization is a well-documented procedure for converting traditionally written laws into expressions of conditions and results without any ambiguity as to the connections among them.[17] Normalized language, as exemplified in Tennessee statutes,[18] is ordinary English coupled with a limited set of words to indicate connections and with typographic conventions to make the language easier to read.[19] The product of normalized interpretation of a legal rule is a list of the elements that constitute the legal rule in an IF-THEN or IF-AND-ONLY-IF-THEN format that uses only "AND" and "OR" to express connections among conditions and "AND" to express connections among results. In this section only a simple illustration of the conversion will be given.[20]

A statute, which the lawyer called the "protective order survival rule," will be used to illustrate the process of conversion to normalized form and some of the ways NLESB helps the lawyer in building the expert system.[21] The enacted statute reads:

> If an order of protection is ordered by a court and either the petitioner or respondent files a complaint for divorce, the order of protection shall remain in effect until the court in which the divorce action lies modifies or dissolves the order.[22]

In normalized interpretation of statutes as developed by Layman E. Allen and others,[23] the lawyer minimizes changes in the language of the

[17] Layman E. Allen & C. Rudy Engholm, *Normalized Legal Drafting and the Query Method*, 29 J. LEGAL EDUC. 380 (1978) [hereinafter *Normalized Legal Drafting*]; *see also Clear Structure*, *supra* note 13, at 507-13; Layman E. Allen & Charles S. Saxon, *Exploring Computer-Aided Generation of Questions for Normalizing Legal Rules*, *in* COMPUTING POWER AND LEGAL LANGUAGE 243, 250-65 (Charles A. Walter ed. 1988).

[18] *See, e.g.*, TENN. CODE ANN. § 33-6-104(a) (1993).

[19] *Reducing Unintended Ambiguity*, *supra* note 1, at 435; *Clear Structure*, *supra* note 13, at 493-505.

[20] Numerous examples of such conversions are available in the literature with detailed analysis of the logic of the interpretations. *See, e.g.*, *Clear Structure*, *supra* note 13, at 457-70, 473-93; *Normalized Legal Drafting*, *supra* note 17, at 384-400.

[21] The rule will be used frequently in this article as an example.

[22] TENN. CODE ANN. § 36-3-603(a) (1991).

[23] *See, e.g.*, *Normalized Legal Drafting*, *supra* note 17, at 393-400. For a fuller treatment, see LAYMAN E. ALLEN & C. RUDY ENGHOLM, MANUAL FOR LEGAL DRAFTSMEN OF

original document in order to minimize the risk of inadvertent change in meaning.[24] In normalized form, with no changes in language except those necessary to show the structure and the negation implicit in the word "until" in propositions 3 and 4, the rule can be expressed as:

IF
(1) An order of protection is ordered by a court,[25] AND
(2) Either the petitioner or respondent files a complaint for divorce, AND
(3) The court in which the divorce action lies has not modified the order, AND
(4) The court in which the divorce action lies has not dissolved the order,
THEN
(5) The order of protection shall remain in effect.[26]

NORMALIZED TEXT (April 21, 1978) (second draft, sponsored by National Science Foundation Research Grant SOC76-10668) [hereinafter MANUAL FOR LEGAL DRAFTSMEN].

[24] *See* MANUAL FOR LEGAL DRAFTSMEN, *supra* note 23, at 23-28.

As is illustrated later in this article at section III. B. & D., the result of minimizing changes is that a number of kinds of changes can profitably be made in the rules to promote consistency of expression, simplicity, and ease of access for the user of the expert system. The kinds of changes which NLESB facilitates in the language illustrate how it could be a useful drafting tool in developing the language of the law as the writer moves from general statement rule-by-rule to thinking more concretely about the law in action as an operating whole and testing its internal relationships.

[25] See discussion of modification of this proposition to eliminate the passive construction and redundancy, *infra* section III. D. 2.

[26] As is the case with other interpretive processes, normalization does not necessarily lead to a single right answer. There are other accurate ways of expressing the statute in normalized form. For example, the rule could be stated as:

IF
(1) An order of protection has been ordered by a court, AND
(2) (A) The petitioner filed a complaint for divorce after the order of protection had been entered, OR
(B) The respondent filed a complaint for divorce after the order of protection had been entered, AND
(3) The court in which the divorce action lies has not modified the order, AND
(4) The court in which the divorce action lies has not dissolved the order,
THEN
(5) The order of protection remains in effect.

Proposition 2 is represented here in separate, OR-connected propositions for the petitioner and the respondent that make the independence of the conditions explicit.

This normalization also illustrates several kinds of changes in the language of the rule that are not required for normalization but may be helpful in expert systems. Such changes, made in the interests of simplicity and efficiency, have turned out to be common in writing expert systems with NLESB. Propositions 1, 2(A) and 2(B) are written in the past tense to reflect the chronological sequence of the conditions. Propositions 2(A) and 2(B) also make explicit the judgment that filing for the divorce takes place after the entry of the order of protection by the other court. Proposition 5 is written in the present tense ("remains") instead of the imperative of the original text ("shall remain"). The latter change is based on the hypothesis that "remains" is simpler than "shall remain," but the

The normalization of the rule is reflected in these features:

1. All relationships among propositions stating conditions or legal results are expressed solely by the words AND, OR, IF, IF AND ONLY IF, and THEN;
2. Those syntax words are used consistently; and
3. Each proposition is complete as a grammatical sentence, is labeled with an identifier that is ordered and unique within the rule, and is indented in an outline format.

NLESB parses the rules on the basis of the proposition's item labels and the assumption of standardized syntax operators. The labeling of each proposition combined with the standardized operators enables NLESB to reason with the rule in either of two ways.[27] If NLESB is given the truth of conditions, it can determine whether particular results follow under the rules. If NLESB is asked about whether a result is true, it can ask about the truth of the conditions from all rules that lead to that result and then determine whether the result follows.

The protective order survival rule is a relatively simple one. Normalization can express clearly, and NLESB can analyze, far more complex and lengthy rules. Appendix A contains an example of the complexity. Because the syntax between propositions is unambiguous, NLESB can reason with the normalized rules regardless of complexity.

B. NLESB Finds Defects in the Normalization

The effort to write consistently is not always successful. Finding inconsistencies of form in large bodies of law can be difficult, especially for the person who composed them, and time consuming as well. This is true even with as simple a format as normalized English. Consequently, the lawyer can benefit from an assistant that can find deviations from the standards for normalization. When the lawyer puts rules in the system, NLESB parses them for defects in the normalization and asks the lawyer to check each one that it finds.

deletion of the deontic operator "shall" could become troublesome for the system builder if, in the future, NLESB were to draw inferences from deontic operators.

[27] Because the item labels, e.g., "(1)," rather than the capitalized syntax terms or item indentation are used by NLESB to analyze the rules, NLESB also could reason with the following expression of the rule:

> If (1) an order of protection is ordered by a court, and (2)(A) the petitioner files a complaint for divorce, or (B) the respondent files a complaint for divorce, and (3) the court in which the divorce action lies does not modify the order, and (4) the court in which the divorce action lies does not dissolve the order, then (5) the order of protection remains in effect.

The authors continue to use the capitalized syntax operators in preparing rules for NLESB and in NLESB's output because prior research shows that, at least with complex rules, the capitalized operators and outline indentation, coupled with the item labeling enable readers to apply the rules more accurately. *See Readability of the Law*, *supra* note 14.

The defects are formal, but they must be found and corrected for an expert system to run well. Finding them is good work for NLESB, but it would be tedious and expensive if it were done by the lawyer. Correcting the defects is good work for the lawyer, because it requires knowledge and expert judgment that only the lawyer has, and because it would require extensive, complex and brittle computer code to handle the subtle complexities that could arise.

NLESB can find several kinds of defects, but it does not detect grammatical incompleteness of propositions. It will spot propositions with the same number, gaps in the proposition labeling, and places where the propositions are not properly nested.[28] For example, in building the protective order system, a proposition in a rule was numbered (4) when it should have been (3). When the lawyer put the rule in the system, NLESB detected the error and reported it to the lawyer, providing an opportunity for revision of the rule in the editor.

Thus, NLESB helps the lawyer with the formal side of the normalization by identifying problems, guiding the lawyer's attention toward the zone of the error and sometimes the kind of error, and giving the lawyer the opportunity to revise the rule to correct its form. Because one of the major difficulties in building an expert system is insuring the accuracy of the system's knowledge, NLESB does not automatically fix such problems. The lawyer, not the modestly intelligent assistant, decides whether the apparent error is in fact an error and what to do about it.

III. IMPROVING THE ARTICULATION AND CONNECTION OF THE RULES

A. Overview

A legal expert system is essentially a dynamic prescriptive legal document. The lawyer who polishes a prescriptive legal document such as a statute or contract looks for ways to improve the expression of its rules, to correct the document, to clarify the connections among the various parts, and to make the document more useful to those who are due to conform their conduct to it. After the lawyer has entered the rules into the system, NLESB can assist the lawyer in the process of polishing so that the expert system is well expressed, accurate, complete, and useable.

[28] "Nesting" refers to the fact that propositions may be subordinate to one another, a relationship indicated by both item labeling and indentation. See, for example, propositions A and B, which are nested in item 2, *supra* note 26.

NLESB is helpful in a number of particulars:

1. Finding connections among rules;[29]
2. Determining the negation of a proposition that contains "not";[30]
3. Finding unexpressed conditions of rules;[31]
4. Eliminating unnecessary variations in language and simplifying, clarifying, and making language more precise;[32]
5. Eliminating legalistic language as obstacles to comprehension;[33]
6. Finding points at which legalistic language may be helpful;[34] and
7. Identifying parts of the expert system that need further explanation for its end users to get the best information it can provide.[35]

B. Finding Connections among Rules and Making Connections Explicit

A legal expert system is more intelligent if it has full information about connections among the rules. As the body of law grows, the likelihood of unrecognized connections among the rules increases, and NLESB becomes more likely to find potential connections that are useful.[36] NLESB asks the lawyer to decide whether such connections are real, whether to make them explicit, and if so, how. If the lawyer decides that the propositions have an identical meaning, the lawyer directs NLESB to unify them. If the lawyer chooses not to unify them, the lawyer may decide that they reveal an implicit connection and write a bridge rule[37] to express the connection.

When the lawyer unifies two propositions, NLESB treats them, despite their different contexts or language, as meaning the same thing.[38] The search for propositions to unify is the keystone of many of the improvements that NLESB helps the lawyer make in the expression of rules and in the expert system's knowledge. Unifying propositions means that, for example, after the truth of an instance of the proposition is established either by the user or by inference from other rules,

[29] *See infra*, section III. B.

[30] *See infra* section III. C.

[31] *See infra* note 43 and accompanying text in section III. B. and section IV. B.

[32] *See infra* section III. D.

[33] *See infra* section III. E. and note 40 and accompanying text.

[34] *See infra* section IV. D.

[35] *See infra* section III. E.

[36] NLESB's unification process assists the lawyer in other ways which are discussed *infra*, in sections III. D. & E.

[37] *See supra* note 9, and *infra* section IV. B.

[38] NLESB accomplishes this by either combining the propositions into a single data unit or by creating an equivalence rule which leaves their verbal differences but records the fact that they mean the same thing. Consequently, the propositions will have the same truth value to the system despite their different contexts. If NLESB has determined one of the unified propositions to be true, false, or unknown, NLESB will treat the unified propositions the same way in all contexts in which they occur.

NLESB infers that all instances of the unified proposition are true in whatever rules they occur.

For example, when rules connect so that a consequent of one is a condition of another, they link together as in the following rules, which are called Rule 1 and Rule 2 for convenience here.

Rule 1
IF AND ONLY IF
(1) The petitioner files a petition for an order of protection AND
(2) IT IS NOT TRUE THAT The petitioner does have a lawyer
THEN
(3) The petitioner files the petition *pro se*.[39]

Rule 2
IF
(1) The petitioner files a petition for an order of protection AND
(2) The petitioner files the petition *pro se*
THEN
(3) The court shall construe the petition liberally in favor of the petitioner.[40]

Under the lawyer's guidance, NLESB would unify proposition 1 in both rules and proposition 3 in Rule 1 with proposition 2 in Rule 2. Once unified, if a user asks whether the court must construe a petition liberally under Rule 2 and the user cannot say whether the petitioner files *pro se*, NLESB can determine whether the petitioner files *pro se* under Rule 1 by asking whether the petitioner has a lawyer.

NLESB works on the assumption that logically related propositions in a body of law are likely to have similar wording. Similar propositions may be related in different ways, e.g., identity of meaning or wording or similarity of meaning due to parallel construction such as in propositions that grant authority and report its execution. The lawyer determines the level of similarity to be sought by setting the level of match that NLESB requires between propositions before proposing them as possibly relevant.

At one point in building the orders of protection expert system, the lawyer told NLESB to consider one rule for unification with all others in the rulebase.[41] NLESB then asked whether to unify exact matches of language automatically. The lawyer decided against automatic unifi-

[39] The "Def. of files pro se" rule reflects the meaning of *pro se*. This is an example of a bridge rule developed by the lawyer to enable the end user to answer a question that involves a term for which the user may not know the meaning - *pro se*.

[40] TCA sec. 36-3-604(a(s5)) (1991) interp of pro se petitions, based on TENN. CODE ANN. § 36-3-604(a) (1991).

[41] The lawyer could have directed NLESB to unify the whole rulebase at one time. The lawyer chose to unify one rule at a time because that gave him greater control and permitted him to focus on particular parts of the rules in the order which seemed most fruitful.

cation because the rulebase had already grown beyond the point at which he could be confident that such identity would imply identical meaning. NLESB then worked through the propositions in the rule chosen by the lawyer to find similar propositions in the rulebase.

Most of the time verbal identity establishes identity of meaning. Propositions can, however, be verbally identical and not have the same meaning.[42] An example of verbal identity where the propositions had different meanings occurred in building the orders of protection system. The proposition "The court holds a hearing" appears in two rules.[43] The rules referred to different hearings so the proposition in one rule could be true, but that would not establish its truth in the other rule; consequently, the propositions could not be unified. The lawyer differentiated the propositions by rewording them as:

> The court holds a hearing on whether to continue the order.
> The court holds a hearing on whether to issue the order.

The new propositions made it easier for the lawyer to unify correctly and for the end user to find relevant rules without looking at irrelevant ones.

At another point in unifying the database, NLESB asked the lawyer whether to unify the following propositions:

> The respondent is the petitioner's spouse.
> The petitioner and respondent are legally married.

After reviewing the propositions in context, the lawyer concluded that they meant the same thing. The lawyer next had to choose the language in which to express the proposition. Because the propositions were conditions in their rules, NLESB was more likely to present them to the end user as questions. The lawyer saw that the first proposition was shorter and directed NLESB to substitute it for the second.

Comparing the propositions also calls to the lawyer's attention situations in which similar language does not have the same meaning despite its being close. For example, NLESB reported:

> Considering unification of the following propositions:
> 1. The petitioner asks to file a petition for an order of protection
> 2. The petitioner seeks to file a petition for an order of protection.

Examination of the context of the two propositions confirmed the law-

[42] The difference in meaning may be based on differences in the meaning of some of the words or there may be different points of reference as illustrated in text. The meaning is determined by interpretation in light of context.

[43] TCA sec. 36-3-605(b(s1)),(a) (1991) extending an OP, based on TENN. CODE ANN. § 36-3-605(b), (a) (1991), and TCA sec. 36-3-605(b(s3)), (d) impl. (1991) OP w/o ex parte, based on TENN. CODE ANN. § 36-3-605(b), (d) (1991), both contain the proposition "The court holds a hearing," but the former rule refers to a hearing held after one of the parties has requested a hearing on extension of the order of protection while the latter rule refers to a hearing held to determine whether to issue an order of protection.

yer's judgment that the difference between "ask" and "seek" meant that the second proposition was broader than the first—if one asks, one seeks, but if one seeks, one may do so by means other than asking. The propositions were not unified, but the lawyer added the bridge rule:

IF
(1) The petitioner asks to file a petition for an order of protection,
THEN
(2) The petitioner seeks to file a petition for an order of protection.

With that rule in place, if an end user of the expert system confirmed that the petitioner asked to file a petition when the user was trying to determine whether the petitioner's situation justified filing a petition, NLESB would infer that the second proposition was true when told that the first was and would partially satisfy the rule regarding the scope of the remedy in which the second proposition is a condition.[44]

When NLESB presents propositions to the lawyer for consideration, the lawyer may recognize implicit conditions that NLESB itself cannot detect. NLESB facilitates this by isolating the proposition so that the lawyer is invited to focus attention on it. For example, while working on unification of the protective order survival rule, the lawyer was shown the proposition "The court in which the divorce action lies does modify the order." The lawyer recognized that the power to modify an order of protection entered by another court was a necessary implication of the proposition. When NLESB did not show the lawyer any proposition that gave the divorce court that power, the lawyer created the following rule to make the system more complete:

IF
(1) A court has entered an order of protection AND
(2) (A) The petitioner files a complaint for divorce OR
(B) The respondent files a complaint for divorce
THEN
(3) The court in which the complaint for divorce is filed may modify the order AND
(4) The court in which the complaint for divorce is filed may dissolve the order.[45]

The expert system then "knew" more than it had from the original rules.

Similarly, propositions that are presented for possible unification may reveal a connection between rules that has not been stated, e.g., that a power exists as a condition for a rule which has as its condition that the power has been exercised. NLESB presented the following propositions to consider for unification:

[44] TENN. CODE ANN. § 36-3-602 (1991).

[45] The rule, IMPL<TCA sec. 36-3-603(a) (1991), is based on TENN. CODE ANN. § 36-3-603(a) (1991).

The court may enter an order of protection.
The court enters an order of protection.

The lawyer recognized that, despite their great similarity, the propositions did not mean the same thing, the second being an apparent exercise of the power granted by the first. The lawyer then elected to read the propositions in context. That revealed the following rule:[46]

```
IF
(1) (A) The court enters an order of protection OR
    (B) The court extends an order of protection
THEN
(2) The court shall assess the petitioner's court costs and attorney's
    fees against the respondent . . . .
```

The lawyer decided that the authority to enter the order of protection was an unexpressed condition of the rule which required assessment of court costs and attorney's fees and inserted it in the rule. The additional condition permitted NLESB to inquire about the conditions for the grant of power if the end user did not know whether the court had the power.[47]

C. Determining the Effect of "not" in Rules

After NLESB confirms that the lawyer has put a law in the system in proper normalized form, NLESB reviews the rule for the word "not." If a proposition contains "not," NLESB asks the lawyer whether the proposition is the negation of the proposition without the "not." For example, NLESB asked:

[46] The rule, TCA sec. 36-3-605(d), 606(d), 608(a), (b), 609 (1991) terms of OP, is based on TENN. CODE ANN. §§ 36-3-605(d), 36-3-606(d), 36-3-608(a), (b), 36-3-609 (1991).

[47] In relevant part, after the addition, the rules were:

```
TCA sec. 36-3-605(b(s3)),(d) impl. (1991) OP w/o ex parte
  IF
  (1) The court sets a date for a hearing AND
  (2) The sheriff serves the order setting the hearing on the respondent AND
  (3) IT IS NOT TRUE THAT the court does issue an ex parte order of protec-
      tion before the hearing AND
  (4) The court holds a hearing AND
  (5) The petitioner has proved the allegation of abuse by a preponderance of the
      evidence
  THEN
  (6) The court may enter an order of protection
TCA sec. 36-3-605(d), 606(d), 608(a),(b), 609 (1991) terms of OP
  IF
  (1)(A)(i)(a) The court may enter an order of protection . . ., AND
         (ii) The court enters an order of protection . . .,
  THEN
  (2) The court shall assess the petitioner's court costs and attorney's fees against
      the respondent . . . .
```

> Is this proposition:
>
> > The court in which the divorce action lies does not modify the order
>
> the negation of this one:
>
> > The court in which the divorce action lies does modify the order? (Y/N)

Because the former proposition is an otherwise identical factual condition, it is the negation of the latter. As a result of the lawyer's confirmation of the negation, NLESB modified the proposition in the rule so that it read:

> IT IS NOT SO THAT The court in which the divorce action lies does modify the order.

Such negated conditions were the most common negation in the orders of protection system.[48]

The word "not" appears in different contexts. The decision that a proposition without "not" is the negation of the proposition with "not" cannot be made mechanically because of variation in what is negated by "not." For example, in building the orders of protection system, NLESB asked:

> Is this proposition:
>
> > The order shall be for not more than one year
>
> the negation of this one:
>
> > The order shall be for more than one year? (Y/N)

The first proposition requires that the order be for one year or less, and the second requires that the order be for over a year. Because the law imposes a maximum duration of one year on a protective order,[49] the second proposition can never be true. These propositions limit the duration of orders, and they are inconsistent,[50] but they are not simple negations of one another. Thus, if NLESB determines that the conditions that would require an order to be for "not more than one year" are not satisfied, it cannot infer that the order shall be for "more than one year."[51]

[48] Among the many negations found were "not less than," TENN. CODE ANN. § 36-3-601(3)(A) (1991), "Nothing," TENN. CODE ANN. § 36-3-603(b) (1991), "not to exceed," TENN. CODE ANN. § 36-3-605(b) (1991), "not limited to," TENN. CODE ANN. § 36-3-606(a) (1991), "No order shall . . .," TENN. CODE ANN. § 36-3-606(d) (1991), "cannot," TENN. CODE ANN. § 36-3-604(a) (1991), and "nor," TENN. CODE ANN. § 36-3-614(a) (1991). Negations were also found in "until," TENN. CODE ANN. § 36-3-603(a) (1991), and "unless," TENN. CODE ANN. § 36-3-605(b) (1991).

[49] TENN. CODE ANN. § 36-3-608(a) (1991).

[50] They are referred to in logic as contraries.

[51] Such a negation could be avoided by writing the first proposition as:

The order shall be valid for a year at most.

Similarly, when NLESB asked

> Is this proposition:
>
> The forms shall not be used except in cases filed under this part
>
> the negation of this one:
>
> The forms shall be used except in cases filed under this part? (Y/N)[52]

the lawyer recognized that the first proposition was a prohibition and that the second proposition made no sense with its apparent imposition of a duty to use the forms only in cases that were not brought for orders of protection. Indeed, the law specifically permitted the petitioner to use other forms.[53]

In its current implementation NLESB does not help directly with the many other forms of negation, such as "no," "nor," etc.[54] An example of a proposition whose negation NLESB did not inquire about is the exclusion rule shown in Appendix A that contained the following proposition:

> No order of protection made under this part shall in any manner affect title to any real property.[55]

However, using NLESB cultivated the lawyer's sensitivity to negation, and NLESB's presentation of propositions for consideration in isolation fostered recognition of the opportunity to simplify the negative formations. Preparing the rules for NLESB also required resolution of such negative syntax terms as "unless" and "until" in a context that fostered minimization and simplification of negative terms as was illustrated above in the normalization of the protective order survival rule.[56]

[52] The rule, TCA sec. 36-3-604(a(s1,2)) (1991) Ct. clerk provides forms, is based on TENN. CODE ANN. § 36-3-604(a) (1991).

[53] TENN. CODE ANN. § 36-3-604(a) (1991) requires the clerk to provide the forms but expressly authorizes the petitioner to present "any legally sufficient petition in whatever form."

[54] *See supra* note 48.

[55] Proposition 6 in the rule, Rule TCA sec. 36-3-606(a), (b), (d), (e) (1991) kinds of relief, based on TENN. CODE ANN. §§ 36-3-606(a), (b), (d), and (e) (1991).

[56] TENN. CODE ANN. § 36-3-603(a) (1991) is an example. It reads:

> If an order of protection is ordered by a court and either the petitioner or respondent files a complaint for divorce, the order of protection shall remain in effect *until* the court in which the divorce action lies modifies or dissolves the order. [Emphasis added.]

As a normalized rule to be put in the expert system it read:

IF
(1) A court has entered an order of protection, AND
(2) (A) The petitioner files a complaint for divorce, OR
(B) The respondent files a complaint for divorce, AND
(3) The court in which the divorce action lies does not modify the order, AND
(4) The court in which the divorce action lies does not dissolve the order,
THEN
(5) The order of protection remains in effect.

Three benefits follow from NLESB knowing that one proposition is the negation of the other. First, posing questions in the affirmative makes the questions simpler to answer and consequently more likely to be answered accurately.[57] For example, if the proposition from the first example in this section were left unchanged and posed as a question, NLESB would ask:

> The court in which the divorce action lies does not modify the order?[58]

The end user might be confused about whether to say yes or no.[59] Without "not" the question is:

> The court in which the divorce action lies does modify the order?

That is a simpler question to answer.

Second, if one proposition is the negation of the other, after extracting "not" the remaining proposition may be subject to unification with other positive propositions.

Third, if two propositions that are unified and one is the negation of the other in operation, once NLESB determines that either proposition is true or false, NLESB knows the truth value of the other.[60] Without being assured that one proposition is the negation of the other, such knowledge could not be inferred directly. The system knows more and is smarter.

D. Making the Knowledge Consistent, Clearer, More Precise, and Simpler for the End User

NLESB was designed to help the lawyer identify related rules (1) to unify propositions that mean the same thing and (2) to formulate rules to connect related rules. As a result of the lawyer's use of NLESB to examine and test the knowledge base,[61] the lawyer improved the rules by making the language in the rules more consistent, clearer, more precise, and simpler. These changes improved both the expression of knowledge built into the system and the knowledge that the system obtains from the end user.

NLESB focused the lawyer's attention on the details of each propo-

[57] *See* Norman E. Gronland, How To Construct Achievement Tests 60-61 (4th ed. 1988), and J.T. Dillon, The Practice of Questioning 26 (1990).

[58] NLESB forms its questions by turning its propositions directly into questions instead of re-ordering or changing the words in the propositions.

[59] *See* Dillon, *supra* note 57, at 26.

[60] NLESB also may determine the truth value of a proposition by the application of another rule or by implication from an IF AND ONLY IF rule.

[61] NLESB was not developed to help the lawyer make the expert system simpler, clearer, or more precise except as the latter might be a by-product of greater consistency achieved by recognizing identity of meaning where expressions varied. For those who are interested in legal writing, however, such improvements are an interesting side effect of setting out to work on the logical structure.

sition and rule it presented. This was accomplished in several different settings:

1. When NLESB presented a proposition alone as in asking whether the lawyer wanted to compare it to similar propositions;
2. When NLESB presented the proposition and a list of others that it found to be similar;
3. When NLESB presented the proposition and another that the lawyer had selected for closer comparison;
4. When NLESB asked whether one proposition was the negation of another;
5. When NLESB used a proposition to pose a question during the hypothetical process;
6. When NLESB explained how it arrived at its answer or why it was asking a question;
7. When NLESB displayed a rule in response to the lawyer's wish to examine rules.

Using NLESB turned out to be much like looking at an object under a microscope. One's full attention is directed to the magnified object, and it is stripped of much or all of its context. Sometimes its context is enriched by showing it with similar propositions. Such focused attention accounted for many of the improvements the lawyer made in the rules while using NLESB.

In particular, using the hypothetical process to test the system cultivated the lawyer's ability to see the expert system from the perspective of its intended users, the end users. This led the lawyer to recognize alternative formulations of the rules that would make the system more responsive to the end user.

1. Consistency

NLESB helped the lawyer achieve greater consistency in the protective orders system by highlighting similarities of language. For example, while reviewing rules to unify, NLESB offered:

Considering unification of the following propositions:

1. The respondent and the petitioner reside jointly in the same dwelling unit.
2. The respondent and the petitioner do reside jointly in the same dwelling unit.

The lawyer's first thought, based on noting that the word "do" was the only difference between them, was that the meanings were identical. After NLESB showed the context of each proposition, the lawyer confirmed the judgment. The lawyer then directed NLESB to substitute the first proposition for the second. That eliminated an inconsequential variation in language that was merely an artifact of the context in which the two propositions originated.[62]

2. *Clarity and Precision*

While legal writing generally is intended to be clear and precise, legal writers do not always write well. One of the special risks of not writing well is ambiguity about who the actor is. By presenting the propositions in isolation or in comparison, NLESB helps the lawyer spot instances in which greater clarity and precision of statement and greater precision of response by the end user can be achieved by making clear who does what to whom.

A simple step toward greater precision is the virtual elimination of the passive voice. Putting the emphasis on the object of an action rather than on the actor may justify the use of passive voice in an essay. In prescriptive writing such a justification may have no place. In the protective order survival rule, NLESB highlighted the proposition,

> An order of protection is ordered by a court.

While the passive voice in the proposition did not obscure the actor because the actor was named, the lawyer eliminated the passive voice as a matter of general practice in building the system. Noting that simply eliminating the passive voice would produce the odd sentence,

> A court has ordered an order of protection,

the lawyer also substituted "issued" for "ordered" so that the proposition read:

> A court has issued an order of protection.

Anthropomorphism can obscure the actor just as the passive voice can. The lawyer concluded that it would be better to write the rules in terms of people's duties than to anthropomorphize things such as notices. Consequently, when NLESB presented the proposition:

> The notice shall advise the respondent that he[63] may be represented by counsel

the lawyer revised it to:

> The court[64] shall advise the respondent in the notice that he may be represented by counsel.

[62] The second proposition's phrase "do reside" resulted from extracting "not" from a proposition that read "do not reside". Rule TCA sec. 36-3-601(4),-602 (1991) Former household member, based on TENN. CODE ANN. §§ 36-3-601(4) and 36-3-602 (1991).

[63] The pronoun had crept in despite an effort to eliminate pronouns in revising the rules into normalized form. Its presence illustrates why the authors may have a future version of NLESB screen for pronouns. The word "this" has similar potential for ambiguity.

[64] Imputing the power to act to the "court" is reification. The lawyer decided, in light of the diversity of people whose acts are the court's, to live with that as simpler than listing them all and less likely to be erroneous than a list.

Considering the proposition further, the lawyer decided that it was simpler and equally accurate to say:

> The court shall state in the notice that the respondent may be represented by counsel.

Finally, the word "counsel" is not the common word that "lawyer" is. Substituting "a lawyer" for "counsel" made NLESB more likely to be responsive to a question asked in common terms:

> The court shall state in the notice that the respondent may be represented by a lawyer.

The lawyer had clarified who had the duty, what the duty was ("to state"), replaced the pronoun "he," thereby eliminating another source of potential ambiguity, and substituted a more common term for "counsel."

Separating compound subjects, predicates, and objects facilitates posing simpler questions and gets more precise, less ambiguous answers. Thus, when NLESB presented the proposition:

> Either the petitioner or respondent files a complaint for divorce

in the protective order survival rule, it was apparent that an end user's affirmative answer to the question generated from that proposition would create ambiguous knowledge for the system: was the filer the petitioner or the respondent? Consequently the lawyer rewrote it as two propositions:

> The petitioner files a complaint for divorce, OR
> The respondent files a complaint for divorce.

When the end user answers a question based on either of those propositions, the knowledge base in the system is unambiguous.

3. *Simplicity for the End Users*

Comparing propositions often leads to recognition that one or both of them can be simplified, regardless of whether they are unified. When NLESB presented the lawyer the following proposition during unification, it was easy to see that it was a legal standard that could be made simpler for the end user to work with:

> the court is in a county with a population of not less than two hundred thousand nor more than eight hundred thousand according to the 1980 federal census or any subsequent federal census.[65]

It could be made simpler because end users might not know the populations of Tennessee counties according to various federal censuses, and

[65] TENN. CODE ANN. § 36-3-601(3)(A) (1991). The authors have found that thoughtful non-lawyers sometimes interpret this language in such a way that a county can fall out of the class described as well as into it. Thus, they might think that if Shelby County is over 800,000 by the census in the year 2000, it would not be included under the rule. Under Tennessee law, Shelby County would remain under the statute if it became over 800,000. *See* Hall v. State, 137 S.W. 500 (Tenn. 1911).

would be more likely to ask questions in terms of counties by name rather than by population ranges and the federal census. Only four counties in fact fit the standard when the lawyer was building it, so the proposition became five propositions:

(a) The court is in Davidson County OR
(b) The court is in Hamilton County OR
(c) The court is in Knox County OR
(d) The court is in Shelby County OR
(e) The court is in any other county with a population of not less than two hundred thousand nor more than eight hundred thousand according to the 1980 federal census or any subsequent federal census.

The last proposition could have been omitted if the system were designed to be used only before the census of the year 2000.[66] Omitting it would further simplify use of the system.

During unification of the protective order survival rule, the lawyer noticed the consequent of the rule, which is written in a common way for statutes:

> The order of protection shall remain in effect.

Because the statute makes the result legally so, the "shall" is unnecessary mandatory language. The lawyer simplified the proposition to:

> The order of protection stays in effect

The language was then more in the form that an end user of the system might ask about. It is not likely that a user would ask, "Shall the order of protection remain in effect?" The change in language is not likely to affect whether NLESB would find the rule for the end user, though it would make the rule easier to recognize.

E. Identifying Parts of the Expert System That Need Further Explanation for Its End Users to Get the Best Information They Can Provide

During the unification process and in testing the system the lawyer is likely to find propositions that need an explanatory "note" to assist end users.[67] NLESB stores such information in notes when the lawyer wants to supplement the rules.

In building the orders of protection system, the lawyer decided at one point to delete legal citations from the text of propositions on the hypothesis that citations would make the text harder to read and that

[66] The time-boundness of the system's reliability would then have to be communicated to the users.

[67] Notes also facilitate system maintenance by enabling the lawyer to store explanations of the rules for use by those who maintain the system later.

the end user would not ask questions in terms of section numbers.[68] Some laws, however, are applicable only in the context of other laws. Consequently, the lawyer decided that it was important for the system to keep track of such limitations as a law's applying only within a particular section.

Section 36-3-605(a) of Tennessee Code Annotated read (in relevant part):

> An immediate and present danger of abuse to the petitioner shall constitute good cause for purposes of this section.

The version of the rule as it was prepared to put into the rulebase was:[69]

> IF
>
> (1) There is an immediate and present danger of abuse to the petitioner
>
> THEN
>
> (2) There is good cause for purposes of sec. 36-3-605.

The lawyer, however, omitted the section reference from the text of the final proposition in the rule and attached the following note to the proposition:

> A special standard for showing good cause is stated in TCA sec. 36-3-605 (1991). It is reflected in rule TCA sec. 36-3-605(a(s2)) (1991) good cause defined.

NLESB marked the proposition to which the note was attached with an asterisk to signal users that a note was available. The end user can recall the note when viewing the rule or engaging in dialogue.

Another situation in which notes proved helpful arose when NLESB reported that the following propositions were candidates for unification:

> The respondent is the petitioner's former spouse.
>
> The respondent was formerly related to the petitioner by marriage.

When the lawyer reviewed the context in which the propositions appeared,[70] it was clear that the propositions did not mean the same thing. The second proposition referred to persons other than the respondent's former spouse such as in-laws and step-children. Consequently the lawyer attached a note to the second proposition explaining that "related by marriage" included in-laws and step-children but not the petitioner's spouse.

[68] Contrast this discussion with that in section IV. D., *infra*, where section citations were kept in the text of the rules.

[69] The rule is named "TCA sec. 36-3-605(a(s2)) (1991) good cause defined".

[70] The rule, TCA sec. 36-3-601(4),-602 (1991) Former family member, is based on TENN. CODE ANN. §§ 36-3-601(4) and 36-3-602 (1991).

IV. TESTING AND REVISING THE EXPERT SYSTEM

A. How the Testing Was Done

As the expert system developed, the lawyer reviewed it to determine whether inaccuracies had crept in during modification of the language of the rules, unification, or the development of bridge rules. The traditional way to determine whether two versions of a set of rules mean the same thing would be to read and compare them. With NLESB the lawyer did that by printing and reviewing the rules in the knowledge base alongside a copy of the original text. That, however, had the usual problems associated with an effort to compare large bodies of text to one another, for example, difficulties in correlating information that was far apart on paper and in keeping details in mind in comparing language.

NLESB helped alleviate these problems. It enabled the lawyer to test the system directly by posing hypotheticals to decide whether the system was solving the problems as the lawyer would. NLESB offered two ways for the lawyer to pose hypotheticals. First, the lawyer could pose a hypothetical by typing a question about legal results and engaging in a dialogue with NLESB in which the lawyer provided the facts by answering questions, found out whether NLESB reached the result that the lawyer expected, and then reviewed how NLESB reached the result. Second, the lawyer could pose hypotheticals by typing one or more facts in a question and then engaging in the dialogue that NLESB initiated based on relevant rules. In the hypothetical process, the lawyer found it fruitful to focus on topics that involved the rules that the lawyer had altered extensively from their original language. Such rules were more likely to have developed errors.

For example, because the protective order system contained several rules authorizing issuance of an order of protection, the lawyer decided to test that part of the system and posed the question:

> Can the court grant a protective order immediately?

NLESB then asked the lawyer whether knowing it was true that,

> The court may immediately issue an ex parte order of protection,[71]

would answer the question. The lawyer answered affirmatively. NLESB next reported that several rules contained the proposition and asked the lawyer to choose one to apply. The lawyer chose one and worked through NLESB's questions to its answer.

Wanting to test NLESB's explanation, the lawyer asked why NLESB posed its first question. NLESB explained that it had asked about a condition in the rule governing issuance of an ex parte order of

[71] The rule, TCA sec. 36-3-605(a(s1)) (1991) issuance of ex parte OP, is based on Tenn. Code Ann. § 36-3-605(a) (1991).

protection and offered to show the rule to the lawyer. The lawyer declined to see the rule once he knew which one it was.

The lawyer wanted to be sure NLESB would invoke a particular rule for determining good cause[72] if the end user did not know whether good cause was shown. The lawyer replied, "Don't know," on NLESB's question as to whether

> The petitioner shows good cause for an order of protection?

NLESB reported that there was a note on that point and offered to show it. The lawyer declined to see it. NLESB then invoked the appropriate rule and posed its question to the lawyer:

> The petitioner shows there is an immediate and present danger of abuse to the petitioner?

After the lawyer answered affirmatively, NLESB reported that:

> The answer to your question:
>
> > Can the court grant a protective order immediately?
>
> is YES.
>
> Explain this answer?

When the lawyer replied affirmatively, NLESB gave the following explanation based on the two rules it used and the facts given by the lawyer:

> The court may immediately issue an ex parte order of protection [TCA sec. 36-3-605(a(s1)) (1991) issuance of ex parte OP (3)] BECAUSE:
>
> > The petitioner files a petition for an order of protection [TCA sec. 36-3-605(a(s1)) (1991) issuance of ex parte OP (1)], and The petitioner shows good cause for an order of protection [IMPL.<TCA sec. 36-3-605(a) (1991) good cause shown (2)] BECAUSE:
> >
> > > The petitioner shows there is an immediate and present danger of abuse to the petitioner [IMPL.<TCA sec. 36-3-605(a) (1991) good cause shown (1)].

B. Testing Reveals the Need for a Bridge Rule

While the focus of the testing was on finding inaccuracies, the testing sometimes revealed the need to connect rules to enable the end user to obtain information that had not been identified during the unification search. An example of this occurred when the lawyer tested the rule regarding General Sessions Court enforcement of contempt citations for violations of an order of protection.[73] NLESB asked whether

> The general sessions court is exercising jurisdiction under this part?

The rule permits punishment for contempt of the General Sessions Court when that court exercises jurisdiction under the orders of protec-

[72] The rule, IMPL.<TCA sec. 36-3-605(a) (1991) good cause shown, is based on TENN. CODE ANN. § 36-3-605(a) (1991).

[73] The rule, TCA sec. 36-3-610(s2,3) (1991) General Sessions courts, is based on TENN. CODE ANN. § 36-3-610 (1991).

tion statutes. The lawyer answered "Don't know" and found that there was no linkage to a rule that gave jurisdiction.[74] Consequently, the end user would be required to make a judgment, without adequate information, about whether the court had jurisdiction. The lawyer then wrote the following bridge rule to give access to information about General Sessions jurisdiction and to link the contempt rule and the jurisdiction rule:

IF
(1) The court has jurisdiction to enter an order of protection, AND
(2) The court is a general sessions court, AND
(3) The court has issued an order of protection, AND
(4) The petitioner has petitioned the court to cite the respondent for contempt,
THEN
(5) The general sessions court is exercising jurisdiction under this part.

With the bridge rule in place, the end user could determine whether the General Sessions Court was exercising jurisdiction under the protective orders law on the basis of simple information. The end user could do so because, while the first question NLESB would ask under the bridge rule (The court has jurisdiction to enter an order of protection?) would probably be beyond the user's ability, the "Don't know" answer would begin questions under the jurisdictional rule that the user might answer more readily.

C. A Side-effect of Testing: Making the Knowledge More Accessible for the End User

Expressing knowledge in the end user's language is one way to make knowledge more accessible to the user. The lawyer often found places to express the rules in the end user's language when NLESB did not bring up the proper rules in response to a question posed as a user might ask it rather than in legal language.

For example, the lawyer asked a question regarding the court clerk's assistance for people preparing their petitions, but NLESB did not pose queries based on the rule the lawyer expected. The lawyer entered as a fact in a "what if" question that "a person asks about a legal remedy." NLESB returned nothing in response to that. The lawyer then asked, "What if the petitioner asks for assistance in filing a petition?", and NLESB began applying the rule which the lawyer expected. The lawyer then reviewed the rule and thought about how an end user might ask the question. The lawyer then added the following condi-

[74] The rule, TCA sec. 36-3-601(3) (1991) Court def., is based on TENN. CODE ANN. § 36-3-601(3) (1991) and gives certain General Sessions courts jurisdiction to issue orders of protection.

tions to the rule so that it would be invoked in response to foreseeable "what if" questions:

> The person asks about a legal remedy or an order of protection.
>
> The person complains about being hurt, abused, or threatened by someone else.

Adding those conditions led the lawyer to realize that "person" was a predictable word in questions about both petitioners and respondents. Consequently the lawyer decided to use the words "petitioner" and "respondent" throughout the system for uniformity, even at points when no petition had been filed. Having decided on the usage of "petitioner" and "respondent," the lawyer then formulated a single rule to establish both terms in the system. The rule replaced two statutory rules defining "petitioner" and "respondent."[75] The lawyer added a note to the rules providing a rationale for use of the terms even though no petition had been filed. As this example illustrates, testing the expert system also may reveal the need for additional conditions, results, and rules.

D. Another Side-effect: Finding Points at which Legalistic Language May Be Helpful

In building the system, the lawyer at one point had decided to eliminate legal citations to make the language simpler for the end user. However, testing the accuracy of the system revealed the desirability of leaving citations in at some points.[76]

The occasional utility of section references was exemplified in a rule that the lawyer tested because it connected to one he had created. The law authorized the court to provide several kinds of relief in an order of protection, but some of them were excluded if the petitioner was defined as a "family or household member" under sections 36-3-601(D) and (E) of the Tennessee Code Annotated.[77] Further analysis for the expert system led the lawyer to conclude the legislature intended that some family and household members, who literally fell within the definitions, should be eligible for the relief despite the exclusion. The lawyer expressed the rule to get that result without including the section citations in the text of the rule.

When the exclusion rule was invoked in testing the system, the lawyer noticed that a user might think that a spouse was a "family member." Without the citation the question would be

> The petitioner is a family or household member?

Consequently, the lawyer restored the citations to deter the end user

[75] TENN. CODE ANN. §§ 36-3-601(5) and (6) (1991).

[76] Contrast this discussion with that at section III. D., *supra* note 68.

[77] TENN. CODE ANN. § 36-3-606(e) (1991) referred expressly to definitions in TENN. CODE ANN. §§ 36-3-601(4)(D) and (E) (1991).

from answering the questions erroneously.[78] The revised proposition as a question was:

> The petitioner is a family or household member as defined in sec. 36-3-601 subsec. 4d or e for purposes of sec. 36-3-606 subsec. e?

Only an informed end user would be likely to answer with anything other than "Don't know." The latter response calls into play the definition rule for the exclusion,[79] with which NLESB begins to elicit information that the end user is likely to be able to provide reliably, for example, with the question:

> The petitioner and respondent are married?

NLESB then determines whether the petitioner is excluded from the relief.

V. CONCLUSION

As is illustrated above, NLESB enables a lawyer to build a legal expert system without requiring the lawyer to be a computer expert. NLESB, like a modestly intelligent assistant, assists the lawyer in finding unexpressed connections between rules. When the lawyer finds such connections and establishes them, the database is more intelligent and, consequently, able to answer more questions. With NLESB the lawyer finds ways to make the rules more accessible to the end user. NLESB can also then assist the end user in obtaining information on the basis of both inferences and key words.

An unanticipated benefit of using NLESB is that it helps the lawyer write rules more clearly. Were NLESB used by a lawyer writing statutes or administrative rules in the first place, it seems likely that NLESB would be useful both to decision making (by identifying perhaps unexpected correlations among laws) and in writing the law more clearly (by encouraging the writer to focus more concretely on the explicit language of the law in small units). Such a use of NLESB could result in laws that are more readily built into legal expert systems.[80]

[78] The lawyer could have used a note to explain before the user answered, but the user might not have checked a note on an apparently simple proposition. The rules to be invoked were also too complex for a simple note.

[79] The rule, TCA sec. 36-3-606(e), 36-3-601(4)(d), (e) (1991) exclusion def., is based on TENN. CODE ANN. §§ 36-3-606(e) and 36-3-601(4)(D) and (E) (1991).

[80] *See Enacted Normalized Statutes*, *supra* note 1, at 389-408.

APPENDIX A

A Sample Rule from the Orders of Protection System

The rule, TCA sec. 36-3-606(a), (b), (d), (e) (1991) kinds of relief, is based on sections 36-3-606(a), (b), (d), and (e) of Tennessee Code Annotated.

```
IF
(1) (A)  The court may enter an order of protection OR
    (B)  The court shall enter an order of protection AND
(2) The court enters an order of protection
THEN
(3) The court may direct the respondent to refrain from abusing or threatening to abuse the petitioner AND
(4) IF AND ONLY IF
    (A)  The court has given the petitioner and the respondent an opportunity to be heard
    THEN
    (B)  IF AND ONLY IF
           (i)   IT IS NOT TRUE THAT the petitioner is a family or household member as defined in sec. 36-3-601 subsec. 4d or e for purposes of sec. 36-3-606 subsec. e
         THEN
          (ii)   The court may grant the petitioner possession of the residence or household to the exclusion of the respondent by evicting the respondent, by restoring possession to the petitioner or by both AND
         (iii)   The court may direct the respondent to provide suitable alternate housing for the petitioner when the respondent is the sole owner or lessee of the residence or household AND
    (C)  The court may award temporary custody of or establish temporary visitation rights with regard to any minor children born to or adopted by the petitioner and respondent AND
    (D)  IF AND ONLY IF
         (i)   (a)  The petition is filed in connection with a petition for paternity OR
               (b)  The petitioner and respondent are legally married, AND
         THEN
         (ii)  The court may award financial support to the petitioner and such persons as the respondent has a duty to support AND
(5) The order may be enforced under chapter 5 of this title AND
(6) No order of protection made under this part shall in any manner affect title to any real property.
```