

Information Technology and Legal Education: Towards 2000

9th & 10th April 1992

Sponsored by

Context Ltd; Linklaters & Paines; Needham & James



Automating the Creation of an Expert System for Giving Advice about Legislative History Research: Project CLEAR's "*Lexpert*"

I. T. Hardy¹

College of William and Mary, Virginia

Keywords: Hypertext - Legal informatics - Decision trees - Expert systems - Legislative history - Databases

Abstract: This article describes techniques that help to automate the creation of an expert system for giving advice about legislative history research. The advice consists of recommendations to use a particular section or index from a particular research aid.

The resulting expert system is a software program called "*Lexpert*." It runs on IBM and compatible personal computers and takes the conceptual form of a decision tree: a researcher chooses one path from several choices; that choice constitutes a "branch" point that leads to a more focused set of choices; the researcher picks one of those choices; this leads to still more focused choices; etc., until the software can give some advice. The decision tree itself is implemented as "hypertext"-a set of files connected by links. Advice can be provided about thirty different research aids and about some 128 different sections within those aids.

The techniques to create *Lexpert* consist of the use of (1) a database to store standardized information about a set of research aids; (2) a series of database reports for writing out that information to disk files in specified ways; (3) software programs for converting these files to a series of new files linked together as a decision tree; (4) a word processing program to alter the format of the decision tree files; and (5) a software program that enables researchers to make use of the decision tree files to answer questions about legal research.

The use of a database to store the information about legislative history research for later output into a hypertext system has many benefits for the creator of such systems, chiefly keeping track of hundreds of links among hundreds of separate files. The database also allows automation of some otherwise tedious chores, in particular the creation of certain types of indexes to the research aids.

Introduction

Lexpert is a software system running on IBM-compatible personal computers² that gives advice about doing federal legislative history research. It contains information about finding the basic documents of legislative history, such as Committee Reports, Prints, and Documents, floor debates in Congress, Hearing transcripts, companion bills, and the like. Based on the user's answers to a few questions, it suggests the use of one of about thirty different research aids such as the Congressional Information Service's *Index to Publications of the United States Congress*, the CCH Congressional Index, the West Company's *U.S. Code, Congressional & Administrative News*, the GPO's *Monthly Catalog of Government Publications*, and others.

In addition to suggesting a particular research aid, *Lexpert* also suggests which specific section of several that the aid might contain would be best used. The *CIS Index to Publications of the United States Congress*, for example, contains a number of different access methods, including indexes or sections organized by bill title, by bill number, by subjects and names, by public law number, by CIS accession number, and so on. *Lexpert* is designed to identify not only the name of a publication overall, like "*CIS Index*," that will satisfy the researcher's need, but also to name the particular section, such as "the Index of Bill Numbers" that the researcher should use within that aid.

Lexpert also offers, though cannot automatically generate, narrative information describing each of the research aids included in the system, plus other information explaining how a researcher would use each section of each included research aid. *Lexpert* does automatically generate certain other non-narrative information such as its own indexes.

Lexpert is an "expert system" in the broad sense of that term: it contains knowledge about legislative history, and can deliver that knowledge to users in different ways, depending on the responses the users make to the software's questions. Expert systems are conventionally thought of as being based on a "knowledge base" to which a computer program called an "inference engine," applies pre-established "rules" in order to make logical deductions from the knowledge.³

Lexpert is constructed in contrast as a decision tree. The "logic" of the tree is pre-determined when the program is created and is therefore "builtin" to the branches of the tree. The "tree" itself comprises just under 200 short text files⁴. Each of these files is linked to certain of the other files. It is this linking mechanism that makes *Lexpert* a hypertext system⁵, and it is the particular arrangement of the links and the content of the linked files that makes it a decision tree.

By relying on an explicit decision tree instead of the "implicit decision tree" of inference-drawing expert systems, *Lexpert* can provide fast response and a simple, reliable program. Because all the information is in files of text, end users can also edit or add to the system, allowing different libraries to tailor *Lexpert*'s knowledge to their own situations.

Although *Lexpert* is an interesting application in its own right, the focus of this paper is on the techniques used to create the system. I have developed these techniques, particularly the use of a database system to store the knowledge and produce the necessary decision trees, during a year of sabbatical research. They worked well with *Lexpert*, and offer two benefits.

First, they allow partially automatic creation of decision tree systems, with potential application to other areas of legal research besides legislative history. The techniques seem general enough, in fact, to apply to other areas of library research and possibly to any knowledge area for which decision trees would be useful.

Second, because a database, not a person, keeps track of the complicated network of links among the "branches" of the decision tree, the database should be capable of writing out the links in any textual

format. The database technique therefore offers the possibility of generating hypertext decision tree software for a variety of different hypertext systems.

In the next section, I will recount efforts with Project CLEAR that have led up to the present work, particularly the work on an earlier hypertext decision tree for legal research called the "Paper Choice."

After that, I will describe the techniques used in the creation of the decision tree in *Lexpert*, first by an example involving used cars, and then with realistic examples from legislative history research.

Finally, I conclude that the techniques described have practical value, both in creating a decision tree for giving advice about legal research, and in managing the many links that connect information together in a decision tree that is implemented as a form of hypertext.

Project History: the Paper Choice

My work on *Lexpert* continues a long-standing project called "Project CLEAR," which stands for Computers in Legal Education: Assistance with Research. My earlier work with Project CLEAR led to the creation of a decision tree-based software system called the Paper Choice. The Paper Choice has been described elsewhere⁶, but briefly stated, it gives advice to law students about doing basic legal research. Information is available about legal digests, encyclopedias, citators, treatises, and the like. As is true with *Lexpert*, the Paper Choice tried to give specific "how to do it" information, not just offer the name of a research aid.

Like *Lexpert*, the Paper Choice is implemented as a hypertext-based decision tree. It differs in the techniques used for its development. I created the Paper Choice decision tree manually: by thinking about the information that would be necessary to pin down a user's research needs, and writing out that information in the form of a progressively refined tree of questions. The ad hoc nature of this process made the development of the system tedious and error-prone. Moreover, the process was complicated by my attempt to add a large number of cross-references to additional information. These additional references took the form of additional hypertext links beyond those needed for the questions in the decision tree and were added to the system in an ad hoc way.

Those who work with hypertext agree that a lot of links to additional or collateral information can be distracting to users and result in the sensation of being "lost in hyperspace."⁷ Though not so widely acknowledged, at least as big a problem with hypertext is that the hypertext developer can get lost in the creation of the system in the first place⁸. A complex set of links makes for a tangled network of interconnections with a near infinite number of paths. This in turn makes for a system of information delivery that is almost impossible to debug: one cannot test out all possible paths a user might select in using the system.

After creating the Paper Choice, I began to consider ways that a hypertext system for legal research advice could be more systematically created and hence less prone to information bugs. It seemed natural and desirable that the computer itself take on some of the task of creating and managing the hundreds and even thousands of links that a non-trivial hypertext system for legal research might exhibit⁹.

Thanks to a generous sabbatical grant from the College of William & Mary, I was in fact able to take the time needed to find the tools and develop the techniques for creating and managing those hypertext links.

Techniques used to create *Lexpert*

The most mechanical part of a hypertext expert system is the decision tree. A decision tree is almost entirely a hierarchical arrangement of questions. More important, a decision tree can be formed by mechanically "inverting" a straightforward list of characteristics.

For example, suppose we want to create a decision tree to help someone choose the right used car. For simplicity, suppose the particular used cars under consideration consist of only four: a Ford Torino, a Nissan Century, a Toyota Corolla, and a Volvo 740GL. Suppose further that the characteristics of each are as shown in Table 1:

Table 1: List of characteristics of four used cars.

Ford Escort

MPG? very good
Age? 5 to 7 years old
Cost? \$2000 to 2500
Warranty? no

Nissan Century

MPG? very good
Age? 3 to 5 years old
Cost? \$0000 to 6500
Warranty? yes

Toyota Corolla

MPG? excellent
Age? 5 to 7 years old
Cost? \$2000to2500
Warranty? no

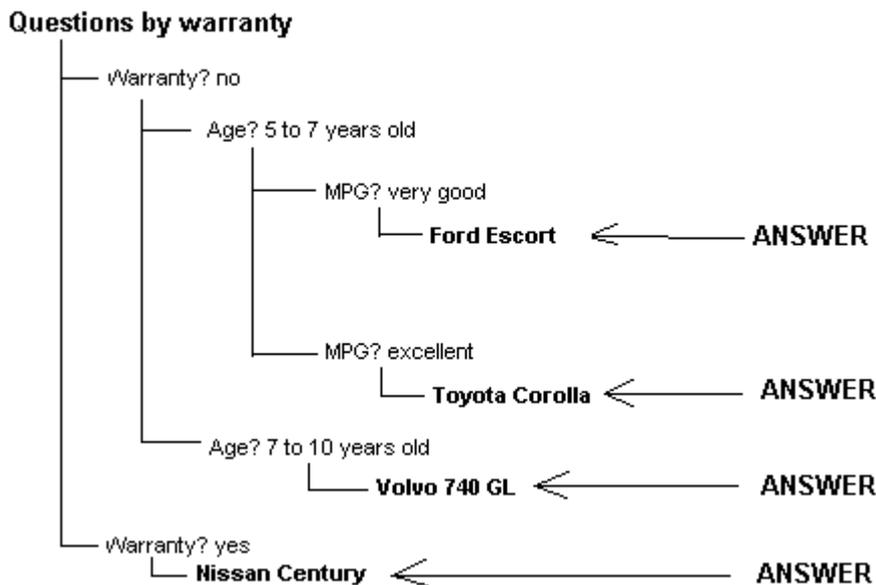
Volvo 740GL

MPG? average
Age? 7 to 0 years old
Cost? \$2000 to 2500
Warranty? no

This list can be viewed as a list of possible answers and their characteristics. The "answers" are: Ford Escort, Nissan Century, Toyota Corolla, and Volvo 740GL. Their characteristics are miles per gallon, model age, likely cost, and existence of a warranty¹⁰.

Once a list like this is created, a software program can convert it into a decision tree. I use a simple decision tree program called "Perm," which stands for "permutations."¹¹ This program accepts files of text in a prescribed format as its input, and produces files of text in the form of a decision tree as its output. The program operates by manipulating strings of text, without regard to the meaning of those strings.

The decision tree so produced can start, or "be rooted at," any of the characteristics. Let us take "Warranty" as the starting characteristic, "Age" as the next, and then "Cost," and "MPG" in that order. The resulting decision tree will then begin by questioning the user about whether a warranty is desired or not; then about the "age" of the car, etc. Such a decision tree is shown in Figure 1:

Figure 1: Decision tree mechanically produced from the list of characteristics In Table 1.

The first decision point leads to two branches: one for which the question of a warranty is answered "No," and one "Yes." This second branch, "Warranty? Yes," leads to the answer "Nissan Century." That is because only one of the four cars carries a warranty, as a glance at Table 1 will show. On the other hand, if the user indicates that a warranty is not required, the branch for "Warranty? No" leads to another branch for "Age." Again, only one car, the Volvo, has the characteristic age "7 to 10 years old," hence, a user who chooses that branch (the second one) will then reach the ~ answer. Finally, if the user chooses an age of "5 to 7 years," two cars will have that characteristic; the decision tree therefore has one more decision branch for the miles-per-gallon question.

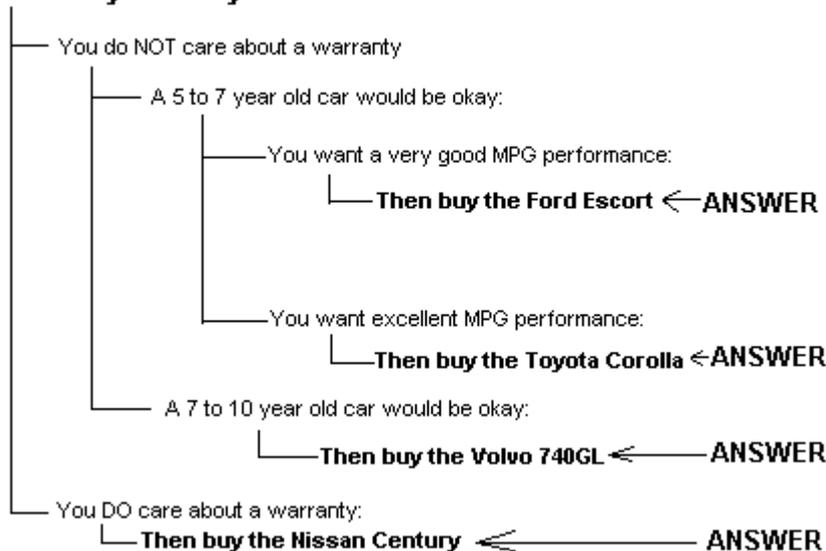
The resulting tree is short partly because the decision tree program eliminates needless questions, and with this trivial example many characteristics will be thus eliminated. With longer lists, of course, fewer characteristics will likely be redundant, so that the resulting decision tree will have more branches.

The point here is that computer software can automatically convert information in the form of a list of "things," each described by a standardized list of characteristics, into a decision tree.

The text in Figure 1 does not appear in the form of questions tailored for use in a decision tree. A different wording would make more sense, either asking questions, or else making statements in a form that users of the system can agree with or not. In Figure 2 is a re-wording of the tree to form statements of agreement:

Figure 2: Decision tree from Figure 1 reworded.

Questions by warranty



The more narrative, statement-of-agreement form shown in Figure 2 can be produced either by running a text-search-and-replace program that edits the mechanically-produced decision tree of Figure 1, or by editing the list of characteristics in Table 1 in the first place so that the resulting text directly produces the decision tree shown in Figure 2.

Techniques applied to legal research

The techniques just described are not specific to used cars or any other area of knowledge. They are usable whenever a decision tree system is desirable for a problem that can be structured as a series of answers, each of which has certain characteristics that define the circumstances under which that answer is appropriate. For a system giving legal research advice about legislative history, it is necessary to replace the list of characteristics describing used cars with a similar list of characteristics describing the tools of legislative history research.

The advantage of starting with a list of characteristics, instead of by trying to create the decision tree right away, is that such lists can be compiled without a great deal of expertise. Simple paper forms can be carried into the library for recording the characteristics of various research aids. As each new batch of forms is transcribed into a computer, the decision tree program can re-generate the decision tree. The system's creator need worry only about getting the characteristics of the research aids recorded properly, not about the correct logic or order of the decision tree.

Table 2 shows, for example, a short list of research aid characteristics. This example deals with one publication: the *CIS Index to the Publications of the United States Congress*.

As with used cars, the choice of characteristics for a legal research aid's section or index is a matter of judgment. I selected three principle characteristics: the range of dates the aid covers; the information a researcher must know to use a particular section of the aid; and the information a researcher must be looking for to make use of the aid's section appropriate.

The range of dates is an obvious and essential characteristic. If a researcher wants a committee print from 1910, it is no good looking in an index of prints that starts with 1970.

The "information a researcher must know" is another way of describing an index's "access point." For example, to use an index of bill numbers, a researcher must already know the number of a bill. The bill number becomes the "information a researcher must know" to use that index.

The third characteristic I chose is "the information a researcher must be looking for." This characteristic refers to the fact that each different index of a research aid is useful for a different purpose. An index of bill numbers in one aid might yield the public law number of the statute that resulted from the bill. In another aid, however, an index of bill numbers might yield the name of the bill's patron in the Congress. An important characteristic in a researcher's selection of a research aid and any particular index within the aid is therefore the information the researcher is looking for

Table 2: Sample list of characteristics for a legal research aid used in legislative history research

The "Legislative Histories" section in the ABSTRACTS volume of the CIS Index to Publications of the U.S. Congress.

- Helps find: the legislative history of a statute
- Section arranged by: Public Law number of the statute
- Date range covered: 1970 to 1983¹²

The "Index of Bill Numbers" in the INDEX volume of the CIS Index to Publications of the U.S. Congress.

- Helps find: the legislative history of a statute
- Section arranged by: the Bill Number
- Date range covered: 1970 to 1983.

The "Index of Titles" in the INDEX volume of the CIS Index to Publications of the U.S. Congress

- Helps find: the Legislative history of a statute
- Section arranged by: the title of the statute
- Date range covered: 1970 to 1983

The "Index of Bill Numbers" in the INDEX volume of the CIS Index to Publications of the U.S. Congress.

- Helps find: the Legislative history of a bill not enacted
- Section arranged by: the Bill Number
- Date range covered: 1970 to current date

The "Index of Subjects and Names" in the INDEX volume of the CIS Index to Publications of the U.S. Congress

- Helps find: the Testimony of a witness at a Congressional Hearing
- Sections arranged by: the name of the witness
- Date range covered: 1970 to current date

For this one publication, the CIS Index to the Publications of the United States Congress, I have selected for illustration four individual sections: the Legislative Histories section of the ABSTRACTS volume, the "Index of Bill Numbers," the "Index of Titles," and the "Index of Subjects and Names." These four sections are useful with three different types of searches: finding the legislative history of a statute, finding the legislative history of a bill not enacted, and finding the testimony of a witness at a Congressional Hearing.

The list of characteristics for this example is shown in Table 2.

Looking ahead to the decision tree that will be produced from this list of characteristics, we would want the name of a research aid's section to become the answer to a researcher's query; the characteristics should then be worded either as questions or statements with which the researcher agrees or disagrees.

Thus the list of characteristics reworded with the decision tree in mind would appear as shown in Table 3.

Table 3: List of characteristics of legal research aids from Table 2 re-worded so that a decision tree produced from the list contains statements of agreement

Use the "Legislative Histories" section in the ABSTRACTS volume of the CIS index to Publications of the U.S. Congress

- You want to find the legislative history of a statute
- You know the Public Law number of the statute
- The statute was enacted between 1970 and 1983

Use the "Index of Bill Numbers" in the INDEX volume of the CIS Index to Publications of the U.S. Congress

- You want to find the legislative history of a statute
- You know the Bill Number of the bill that was enacted
- The statute was enacted between 1970 and 1983.

Use the "Index of Titles" in the INDEX volume of the CIS Index to Publications of the U.S Congress.

- You want to find the Legislative history of a statute
- You know the title of the statute
- The statute was enacted between 1970 and 1983

Use the "Index of Bill Numbers" in the INDEX volume of the CIS index to Publications of the U.S. Congress

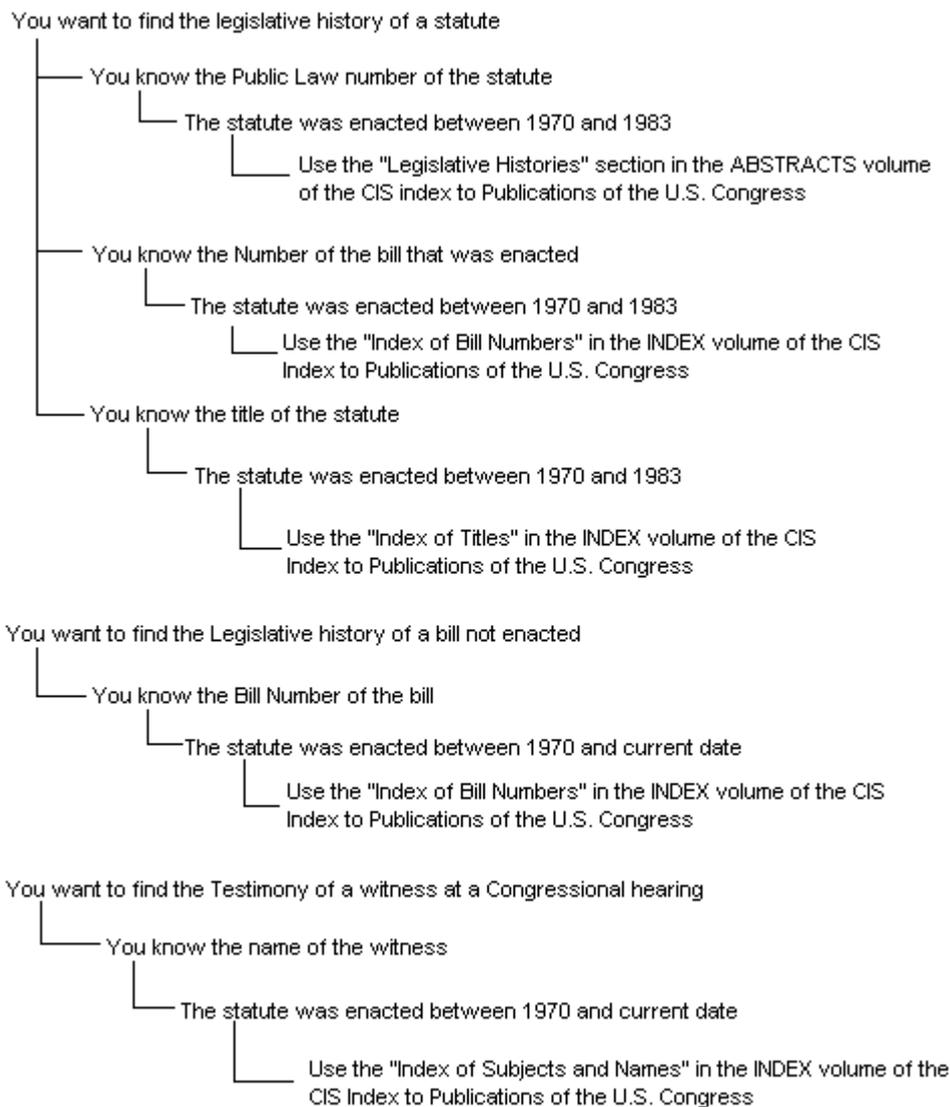
- You want to find the Legislative history of a bill not enacted
- You know the Bill Number of the bill
- The statute was enacted between 1970 and current date

Use the "Index of Subsects and Names" in the INDEX volume of the CIS Index to Publications of the U.S. Congress

- You want to find the Testimony of a witness at a Congressional Hearing
- You know the name of the witness
- The statute was enacted between 1970 and current date

When the decision tree program inverts the list of characteristics in Table 3 to form a decision tree,¹³ the resulting tree comes out as shown schematically in Figure 3.

Figure 3: Decision tree mechanically produced from the list of characteristics in Table 3.



With the used car example, I allowed the program to eliminate redundant branches of the tree; here, however, I have left in the redundancies so that the branches shown will correspond to the characteristics listed in Table 3.

The close correspondence between Table 3 and Figure 3 helps show that the process of creating a decision tree from a list of characteristics is mechanical. But at the same time the simplicity of the example may mask all that a decision tree creating program does. For one thing, as noted, it eliminates redundant questions. In the short excerpt of Figure 3, all of the date range questions are redundant because the advice given would be the same with or without a qualifying date range. If this figure were the final product and not an illustration, the unnecessary questions would have been omitted.

For another, the mechanical nature of the conversion from a list of characteristics to a decision tree ensures that repeated re-generations of the tree can be made without fear of logical errors in the tree structure. When the number of items in the list of characteristics grows, so does the complexity of the tree and the consequent benefit of automation.

Practical problems

The mechanism just described works well in generating decision trees from short lists of

characteristics that have been created "by hand," that is, in an ad hoc way. The technique begins to break down if longer lists of characteristics have to be created by hand. The problem lies not in the decision tree program¹⁴, but rather in practicalities: with long lists of characteristics it is difficult to ensure that all characteristics are listed where they should be and not where they are inappropriate, and that all are spelled and capitalized the same. If these conditions are violated, a program like Perm that depends on comparing strings of text cannot process the list properly.

Although consistent spelling and capitalization sound like trivial requirements, they are not. The effort involved in creating a list of characteristics is large because it requires a slow buildup of answers and characteristics that constantly changes. My experience was that with fifty or more answers (or "pieces of advice"), each with four or five characteristics,¹⁵ some mechanism to control vocabulary and spelling is essential to ensure consistency.

A database approach

A database system is the way I chose to provide that control. A database can be set up so that each "answer" and its associated characteristics form the fields of a single record. By sorting on different characteristics one is able to spot alternative spellings or capitalizations for what should be the same word or phrase.

For the *Lexpert* project, I chose the Borland company's "Paradox" database product because it was available on a site license at my university.¹⁶

I set up a record in the database for each piece of specific advice. For example, there is a record for "Use the 'Index of Titles' from the *CIS Index to the Publications of the U.S. Congress*." With this approach - creating one record for each section or index within an overall research aid-many records will refer to the same publication; the *CIS Index*, for example, has several sections and has changed its indexing coverage overtime. Each section and change of coverage necessitates a separate record in the database.

Rather than repeat the full name of the each research aid in each of these records, I gave each overall publication a code number, called a "publication number."

For example, the *CIS Index to Publications of the U.S. Congress* is assigned the number "12001;"¹⁷ the CCH Congressional Index has the number "12003;" and so on.

In a separate database I then put the full name of the *CIS Index*, along with the full names of all other publications research aids referred to throughout *Lexpert*, arranged by publication number. The database can then maintain the relationship between the two databases¹⁸ by means of a common field, which is here the publication number field.

Specifically, when the database is instructed to write out a report from the "advice" database, it can use the publication numbers listed there to pull each research aid's full name out of the publication database.

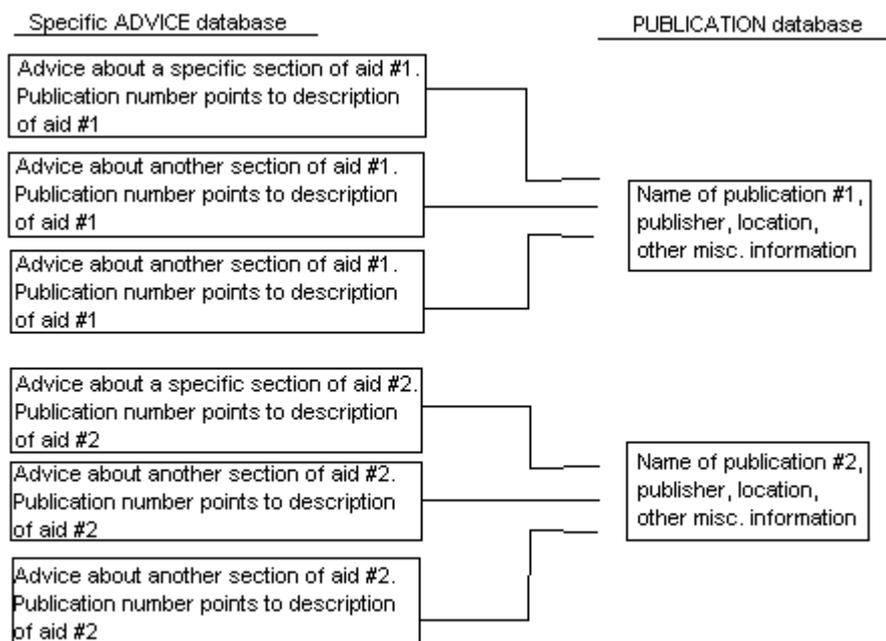
I also use the latter database to store a textual description of where each research aid is located in the William & Mary law library. The use for this information will be described later.

It also turned out to be necessary to give each record in the database of specific advice a number. Again, I used arbitrary numbers made up as I entered more and more information into the advice database; but to keep these record numbers distinct from the publication database record numbers, I preceded each advice record with the letter "R."¹⁹ The record numbers therefore look like this:

"R1001", "R1510", etc.

Figure 4 shows the structure of the database. Note that records in the specific "advice" database have a many-to-one relationship to records in the "publications" database:

Figure 4: Relationship of the "advice" database to the "publications" database.



Fields in the database

The fields in the advice database encapsulate the characteristics of date range, what a researcher must know, and what a researcher must be looking for, in order to make a specific section of a research aid appropriate for the research task. Each record in the Paradox database has a field called "DATES," showing the range of dates (as a string of text) for which the aid is appropriate.

Following the result of some earlier investigations I undertook in the "*Paper Choice*" project²⁰, I used the term "input" to refer to what researchers must know to use a particular index. Similarly I use "output" to refer to whatever the index yields as the result of its use.

The "advice" database therefore has fields called "INPUT" and "OUTPUT" as well as "DATES."

As I collected more entries in the database, I began to see that only some of the answers related to research aids that featured legislative materials as their "output;" others featured Presidential materials as their output. Of those that dealt with legislative branch materials, some were concerned with legislative history because they were indexes to Hearings, debates, Committee Reports, and so on. But others were of a more miscellaneous nature: indexes of Senate Committee members, or indexes that yield as their output the address of each Representative in Congress.

I therefore began to classify each answer in terms of its general area. I termed these areas the research aid's "jurisdiction," and thus set up a field in each Paradox record called "JURISDICTION."

The resulting fields in the advice database are shown in Figure 5, which contains five records as an example. The "SEC_NAME" field will ultimately end up as the specific advice to be given to researchers.

Only two different "JURISDICTION" entries are shown; one for "Legislative history materials:" and the other for "legislative materials." The term "legislative materials" will not be readily distinguishable from "legislative history materials" for most readers²¹. I use the former term to characterize information that researchers would likely want to find apart from any effort to compile a legislative history. Typically this would be the sorts of information referred to previously: names and addresses of Members of Congress, and so on.

As with any scheme, there will be cases difficult to classify, and Figure 5 shows one such case: the finding of the testimony of a witness at a Hearing. I classify this research task as relating to "legislative materials," although one could consider it to relate to "legislative history materials."

Figure 5: Five-record excerpt from database of specific advice.

REC_NUMBER: R1001
 SEC_NAME: "Legislative Histories", ABSTRACTS volume
 PUB_NUMBER: 12001
 OUTPUT: Legislative history of a statute
 DATES: 1970 to 1983
 INPUT: the Public Law number of the statute
 JURISDICT: Legislative history materials

REC_NUMBER: R1002
 SEC_NAME: "Index of Bill Numbers", INDEX volume
 PUB_NUMBER: 12001
 OUTPUT: Legislative history of a statute
 DATES: 1970 to 1983
 INPUT: the Bill Number of the bill that was enacted
 JURISDICT: Legislative history materials

REC_NUMBER: R1004
 SEC_NAME: "Index of Titles", INDEX volume
 PUB_NUMBER: 12001
 OUTPUT: Legislative history of a statute
 DATES: 1970 to 1983
 INPUT: the title of the statute
 JURISDICT: Legislative history materials

REC_NUMBER: R1009
 SEC_NAME: "Index of Bill Numbers", INDEX volume
 PUB_NUMBER: 12001
 OUTPUT: Legislative history of a bill not enacted
 DATES: 1970 to current date
 INPUT: the Bill Number of the bill
 JURISDICT: Legislative history materials

REC_NUMBER: Ri 010
 SEC_NAME: "Index of Subjects and Names," INDEX volume
 PUB_NUMBER: 12001
 OUTPUT: Testimony of a witness at a Congressional Hearing
 DATES: 1970 to current date
 INPUT: the name of the witness
 JURISDICT: Legislative materials

The second or "publication" database contains information about each research aid overall. Five

records are excerpted for illustration in Figure 6.

Note that each record in the database of specific advice (Figure 5) has a field labelled "PUB_NUMBER." This number is duplicated in the database of publications (Figure 6). These numbers constitute the linking mechanism that enables the database system to match a record of specific advice to its "parent" record, the record that describes the research aid overall of which the specific section is a part.

Figure 6: Five-record excerpt from the publication database.

PUB_NUMBER: 12001
 PUB_NAME: CIS/Annual Index to Publications of the U.S. Congress
 PUB_DATE: 1970 to date
 PUBLISHER: Congressional Information Service
 WHERE_IS_IT: First floor, microforms room; on right from lobby entrance

PUB_NUMBER: 12002
 PUB_NAME: U.S. Code, Congressional & Administrative News
 PUB_DATE: 1941 to date
 PUBLISHER: West Publishing Co.
 WHERE_IS_IT: First floor, back row of stacks

PUB_NUMBER: 12003
 PUB_NAME: CCH Congressional Index
 PUB_DATE: 1937 to date
 PUBLISHER: Commerce Clearing House (CCH)
 WHERE_IS_IT: First floor, Reference Room; left row of stacks toward the back

PUB_NUMBER: 12004
 PUB_NAME: CIS U.S. Serial Set Index
 PUB_DATE: 1889-1969
 PUBLISHER: Congressional Information Service
 WHERE_IS_IT: Swem library - ask at the Reference desk

PUB_NUMBER: 12010
 PUB_NAME: Digest of Public General Bills and Resolutions
 PUB_DATE: 1936-date
 PUBLISHER:
 WHERE_IS_IT: First floor, Reference Room. left row of stacks toward the back

Reports to disk create the necessary files

Once the records are entered into the two databases, the database can be instructed to write out "reports" to disk in a form usable by the decision tree-creating program. The first report is simply the list of characteristics: the "answers" and the circumstances under which those answers are appropriate.

A schematic representation of the template²² or outline for this report is shown in Figure 7. Literals are shown in roman type; variable data, filled in from the various records' fields when the report is created, are shown as field names in square brackets in italics:

Figure 7: Report Template for Paradox to Produce a List of Characteristics.

Use the *[Pub_tabl ->PUB_NAME]*. You want this section:
 ANSWER: *[SEC_NAME]*. For more information, see <mor
[REC_NUMBER]>.
 INPU: *[INPUT]*
 OUTP: *[OUTPUT]*
 JURI: *[JURISDICT]*
 DATE: *[DATES]*

Notice the line "For more information see <mor*[REC_NUMBER]*>." The expression in <angle brackets> will become a link to a file name in the final, decision tree form of the *Lexpert* system. The use of these angle brackets will be more fully described below. The use of this particular ifie name "mor..." is explained below.

The expression "*[Pub_tabl->PUB_NAME]*" is the Paradox database syntax to indicate that the field to fill in on the report is called "PUB_NAME" and that it resides in a separate database, called "Pub_tabl." As noted, the field called "PUB_NUMBER" has been elsewhere specified as the link between the two databases.

For convenience in manipulating the report resulting from the template, each line in the template is prefaced with an abbreviated description of the contents of the line; these descriptions-"INPU:", "OUTP:", etc.-are nearly identical to the database field names, though that is simply a convenience. These abbreviations will be used later by a word processing macro to help determine formatting, then deleted.

When the database generates the report described by this template, it looks as shown in Figure 8 (limited here to an excerpt of five records). Note that this list of characteristics will serve the same function as the list of used car characteristics shown in Table 1.

Figure 8: Five-record list of characteristics from the database report created from the template in Figure 7.

Use the CIS/Annual index to Publications or the U.S. Congress
 ANSWER: You want this section:
 "Legislative Histories," ABSTRACTS volume. For more
 information, see <morR1001>.
 INPU: the Public Law number of the statute
 OUTP: Legislative history of a statute
 JURI: Legislative history materials
 DATE: 1970 to1983

Use the CIS/Annual Index to Publications of the U.S. Congress
 ANSWER: You want this section:
 "Index of Bill Numbers," INDEX volume. For more Information,
 see <morR1002>.
 INPU: the Bill Number of the bill that was enacted
 OUTP: Legislative history of a statute
 JURI: Legislative history materials
 DATE: 1970 to1983

Use the CIS/Annual Index to Publications of the U.S. Congress
 ANSWER: You want this section:
 "Index of Titles," INDEX volume. For more information, see
 <morR1004>.

INPU: the title of the statute
 OUTP: Legislative history of a statute
 JURI: Legislative history materials
 DATE: 1970 to 1983

Use the CIS/Annual Index to Publications of the U.S. Congress

ANSWER: You want this section:
 "Index of Bill Numbers," INDEX volume, For more information,
 see <morR1009>.

INPU: the Bill Number of the bill
 OUTP: Legislative history of a bill not enacted
 JURI: Legislative history materials
 DATE: 1970 to current date

Use the CIS/Annual Index to Publications of the U.S. Congress

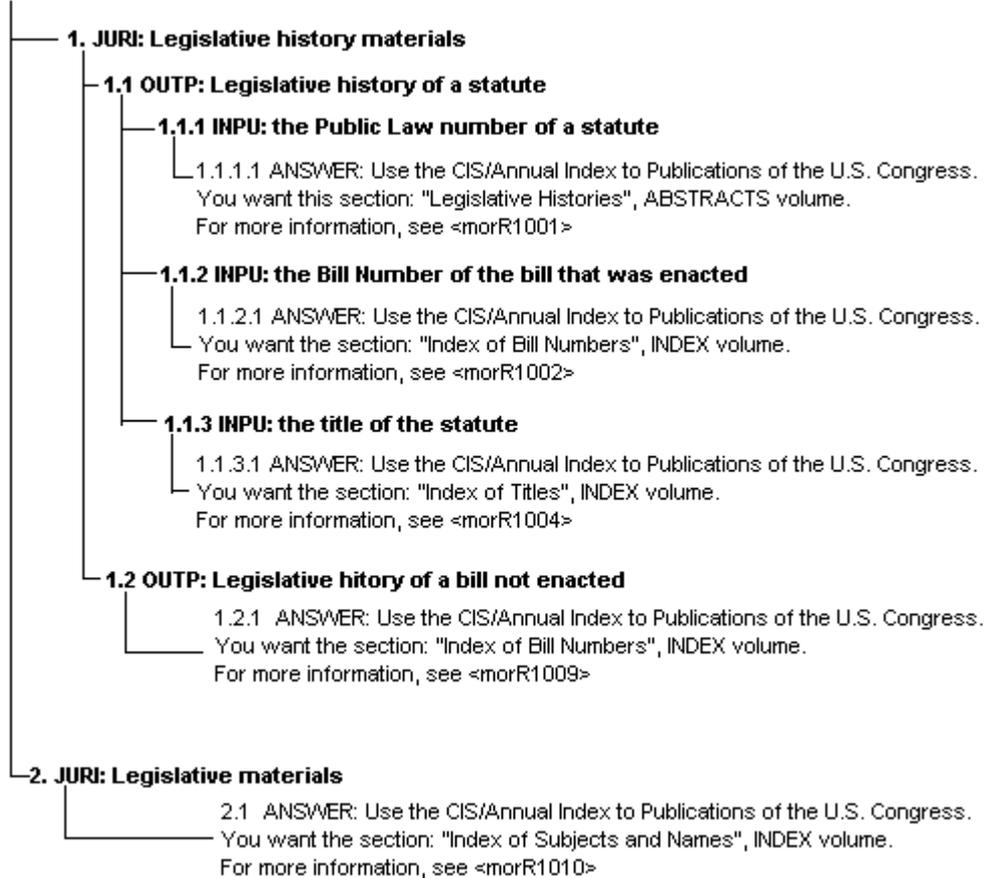
ANSWER: You want this section:
 "index of Subjects and Names," INDEX volume. For more
 information, see <morR1010>.

INPU: the name of the witness
 OUTP: Testimony of a witness at a Congressional Hearing
 JURI: Legislative materials
 DATE: 1970 to current date

:

When the decision tree-creating program reads the information shown in Figure 8, plus some other necessary information²³ it produces the file shown in Figure 9.

Figure 9: Decision tree produced from the file shown in Figure 8.

Decision Tree

Recall that whenever any questions would be redundant, the decision tree program automatically eliminates them. As has often been the case with these short examples, several questions are unnecessary; here I have chosen to let the program omit them. Again, I have added lines to the figure to emphasize the branching nature of the tree; the actual disk file contains indenting to represent the levels of branching in the tree.

The military-style numbering in italics is not part of the program's output and does not appear in *Lexpert* itself; rather I have added it here as a reference for the reader to use in comparisons with Figures 10 through 13, below.

File splitting and linking

What remains is some mechanism for reading and presenting the decision tree to end-users for use. Many outline (or "idea") processors can read the indented file format that the decision tree program produces and present the questions in the appropriate order, but for distribution to end users, it is not suitable to require the purchase of an entire outlining program²⁴.

One method for making a distributable version of the decision tree is to use a "file splitting" program to break the tree into small text files, and then use a corresponding program designed to read and jump among the files in response to a user's choices. Both a file splitting program and a file reading program are available from the author of the decision tree program for just this purpose²⁵.

The "splitter" program produces plain ASCII text files, one file for each set of decision tree branches that grow at the same level from a common node. The hypertext links among these files take the form of file names surrounded by angle brackets like <this>. Each file by default is created with a title line that is the line of the previous file from which this file represents a descendent, that is, a "linked-to" topic.

Figures 10 through 13 show the first four such files, produced mechanically from the decision tree file shown above in Figure 9. These files are "linked" together by virtue of the file names in angle brackets.

On a computer screen, the "linking" means that with a single key press, the file named in angle brackets can be brought to the screen; then another; then another; etc. With a different key press, these files can be viewed in reverse order: first the file that was viewed just previous to the current one; then the file just previous to that one; then the one previous to that one; etc. This is standard hypertext technology applied to the execution of decision trees.

First decision tree

1. JURI: Legislative history materials <TREE02>
2. JURI: Legislative materials <TREE08>

Figure 10: File "TREE001" before editing.

In the files shown in Figures 10 through 13, for example, readers can follow one particular chain of links to its conclusion. I have added italicized military numbering once again so that readers can determine from where in the single decision tree file of Figure 9 each of the "split out" files of Figures 10 through 13 has come.

For illustration, only the link that appears first in each file is shown here. The first file is "TREE001" (Figure 10), and the first link in that file is to <TREE002>. File "TREE002" is shown in Figure 11.

1. JURI: Legislative history materials

- 1.1 OUP: Legislative history of a statute <TREE03>
- 1.2 OUP: Legislative history of a bill not enacted <TREE07>

Figure 11: File "TREE002" before editing.

Looking at file "TREE002" (Figure 11) shows that the first link it contains is to another file named "TREE003," shown in Figure 12. Compare the numbering in each figure to keep track.

Figure 12 : File "TREE003" before editing.

1.1 OUP: Legislative history of a statute

- 1.1.1 INPU: the Public Law number of the statute <TREE04>
- 1.1.2 INPU: the Bill Number of the bill that was enacted <TREE05>
- 1.1.3 INPU: the title of the statute <TREE06>

The first link in "TREE003" (Figure 12) is to a file named "TREE004," shown in Figure 13.

Figure 13: File "TREE004" before editing.

1.1.1 INPU: the Public Law number of the statute

- 1.1.1.1 ANSWER: Use the CIS/Annual Index to Publications of the U.S. Congress. You want this section: "Legislative Histories", ABSTRACTS volume. For more information, see <morR1001>

"TREE004" in Figure 13 contains the answer to the research query. Had other links been followed than the first in each file, a different set of files would have been brought to the screen. This would represent a different path through the decision tree, or a different set of responses from the user and hence the description of a different research problem. The result would, of course, have been a different answer.

This set of files, produced from the database with essentially no human intervention, can be read by the hypertext reading program and used to answer research questions; it thus forms the heart of the *Lexpert* system. The key point here is that the complex linking of branches in the decision tree has been managed by software.

Graphic design touch-up with word processing macros

Though the file displays shown in Figures 10 through 13 are functional, they are visually unappealing. But because they all share a rigidly standard format, applied by the file splitting program, a text search-and-replace program can read them, one at a time, and improve their appearance.

This cosmetic surgery process can be arbitrarily complex, but for good appearance I wrote a word processing macro²⁶ that finds the four-letter labels on each line, deletes them, inserts a more expanded question in place of the default heading supplied by the file split utility, replaces the row of "equal" signs with a single line, makes lower case the file names that appear in angle brackets, and adds small "bullet" characters alongside each question.

When the macro sees the four-letter abbreviations "INPU," "OUTP", etc., or the word "ANSWER," it makes the adjustments shown in Table 4.

Table 4: Editing changes in files effected by word processing macro.

This key word	is changed by a word processing macro this way
ANSWER:	Changes "ANSWER" to "SHORT ANSWER-TRY THIS:" surrounded with heavy lines to draw attention to the fact that this is indeed what the user was looking for. ²⁷
INPU:	Deletes "INPU:". Inserts at top of file this phrase: "What information do you have already:"
OUTP:	Deletes "OUTP:". Inserts at top of file this phrase: "What specifically are you looking for:"
JURI:	Deletes "JURI:". Inserts at top of file this phrase: "What type of material are you researching:"
DATE:	Deletes "DATE:". Inserts at top of file this phrase: "Within what range of dates are you researching? (Dates shown may overlap because the coverage of different research tools overlaps; when that is true, the easiest tools to use are listed first):"

In addition to the editing shown in Table 4, the word processing macro adjusts the file's margins and spacing for a consistent and pleasing appearance when displayed. Readers unfamiliar with hypertext may not realize that successful hypertext systems require the system's author to pay a great deal of attention to graphic design (text layout, spacing, margins, and so on).²⁸ A tremendous advantage of using a system of plain ASCII text files for hypertext is that familiar tools like word processors can be used for just that purpose and can automate much of the formatting necessary.

The resulting files "TREE001," "TREE002," etc., look like the files shown in Figures 14 through 17 after the word processing macro has edited them. Compare these "after" diagrams with the "before" diagrams of Figures 10 through 13.

Figure 14: File "TREE001" after editing.

What type of material are you researching;

- Legislative history materials <tree002>
- Legislative materials <tree008>

ANSWER: Try this research aid	CIS/ANNUAL INDEX TO PUBLICATIONS OF THE U.S. CONGRESS
Turn to this particular section	LEGISLATIVE HISTORIES ABSTRACTS VOLUME
To recap your question:	<recR1001>
Is this aid available here?	<whe12001>
How do you use it?	<howR1001>
What is this aid generally?	<wha12001>

Press CTRL + HOME to return to the beginning of Lexpert and start all over

These "rec" files are generated automatically as reports from the database. Recall that each piece of specific advice is stored as a single record in the database. The advice is the name of a section from a publication overall. The remainder of the record associated with that advice is the collection of characteristics that define the circumstances under which the advice is appropriate: the "input," "output," "jurisdiction," and "date."

These characteristics become the path that users follow through the decision tree. Because all of these characteristics are contained in a single record with a unique record number, they are "known" ahead of time by the database. It is therefore a trivial matter to set up a database report template that writes out to a disk file a report consisting of screens like that in Figure 19 for each record in the database.³⁰

Thus the files shown in Figures 18 and 19 relate to the record in the advice database that carries the record number "R1001." The file named "morR1001" in Figure 18 therefore stands for "more information about record number R1001." Similarly, the file "recR1001" in Figure 18 stands for "a recapitulation of the information from record number R1001." Readers can turn back to Figure 5 and see record R1001; it is the first record illustrated there.

Figure 18 contains a line that says "How do you use it?" This line ends with a link to a file named "howR1001." Again, this file can be created before or after the creation of the decision tree because its name and contents were determined as soon as record number R1001 was added to the database. Its name stands for "how to use the section and research aid identified in record number R1001." The file for "howR1001" appears in Figure 20.

Figure 20: The "how to use" it file "howR100" giving information on how to use the research aid identified on the answer screen shown in Figure 18.

To find: Legislative history of a statute
dated: 1970 to 1983
knowing: the Public Law number of the statute
Use--> CIS/ANNUAL INDEX TO PUBLICATIONS OF THE U.S. CONGRESS
Sectn--> "Legisiative Histories", ABSTRACTS volume

Find the ABSTRACT volumes that cover the year of your statute. Turn toward the back of that volume and look for a heading that says "Legisiative Histories." On more recent volumes that phrase appears as a running head at the tops of pages; on earlier volumes, you will just see a Public Law number as a running head.

Find the entry for your statute. Under the entry you will find a list of the numbers of all related Bills (that did not pass), plus Congressional Reports, Hearings, Committee prints, etc., plus citations to the Congressional Record where floor debates are recorded. You will also find CS "Accession Numbers" for many of these Reports, Prints, etc. (but

not for the floor debates). These numbers will look like "H321-1" or "S491-12" or something similar.

In the main part of the ABSTRACTS volume, there are entries arranged by these CIS accession numbers. Look up the numbers

There you will find an abstract of the contents of each document. If individual witnesses testified before Congress at a Hearing, their names will be listed as well.

Because this "how to do it," information is largely narrative and the Paradox database does not handle large text fields³¹, I prepared the descriptions with a word processor. These descriptions are stored in separate files, one for each record, outside of the database. Still, a database report template does generate all the title lines at the top, thereby ensuring accuracy and consistency from file to file.

Notice that Figure 18 contains two links that do not refer to record "R1001." These are the links for "wha" and "whe," appearing in the Figure as "<whe12001>" and "<wha12001>". These links lead to information about "where" in the library a research aid is located, and "what" in general the research aid is for.

Both of these links relate to information about a research aid publication overall, not about a specific section or index of an aid. The links are therefore keyed not to record numbers in the database of specific advice, but rather to the publications database. The file name "whe12001" stands for "where is the research aid identified in record number 12001 of the publications database?" Likewise, the file name "wha12001" stands for "what is the research aid identified in record 12001?"

Readers can check the database excerpt in Figure 5 to see that record number R1001 contains a "PUB_NUMBER" of 12001; that publication number cross references a record with the same number in the publications database. Refer to the excerpt of the publications database in Figure 6 to see that record number 42001 is for the CIS/Annual Index to Publications of the U.S. Congress. The last field in that record, labelled, "WHERE_IS_IT," contains the information-that I added weeks earlier to the database-that the CIS/Index is located in the William & Mary law library on the "First floor, microforms room; on right from lobby entrance."

When the *Lexpert* system is generated from the database, a report template writes out a file named "whe12001" with the contents of the "WHERE_IS_IT" field. Similar files are written out for each of the other records in the publications database. File "whe12001" is shown in Figure 21.

Figure 21: The "where" file "whe12001" giving the physical location of the research aid Identified on the answer screen shown in Figure 18.

WHERE: CIS/Annual Index to Publications of the U.S. Congress
First floor, microforms room; on right from lobby entrance

The last link in Figure 18 tells users, who have just been advised to use a particular research aid, that they can learn "What is this aid generally" by pursuing a link denoted "<wha12001>". Like the "how" link, this link connects the user to a file whose narrative text was "hand written," but whose title lines were automatically generated by a report template from within the database.

Figure 22 shows the "wha" file for the answer screen shown in Figure 18 - that is, for the overall publication CIS/Annual Index to Publications of the U.S. Congress.

Figure 22: The "what" file "wha12001" giving general information about the research aid identified on the answer screen shown in Figure 18.

WHAT: CIS/Annual Index to Publications of the U.S. Congress

The Congressional Information Service (CIS) publishes a number of materials relating to the U.S. Congress. The primary on-going index is the CIS/INDEX, which when bound is called the CIS/Annual. This publication began coverage in 1970.

The INDEX Indexes Congressional Reports, Hearings, Prints, and Documents. The INDEX is accompanied by ABSTRACTS, which summarize the contents of each of the publications indexed.

From 1970 to 1983, these publications were cumulated annually into just those two volumes, the CIS/Annual: INDEX and the CIS/Annual:ABSTRACTS. In those years, the ABSTRACTS volume included a section toward the back titled "Legislative Histories," which gave the history of publications relating to bills and statutes.

Since 1984, the Legislative History section has been pulled out and separately bound in a volume titled CIS/Annual:LEGISLATIVE HISTORIES.

Index files produced from the database

As the preceding tables and figures show, the use of a database and other tools can facilitate the creation of a decision tree for giving advice about legal research. Even those files that had to be "hand written" gained a standard format and appearance from reports written out by the database.

The value of having the bulk of the substantive information about legal research contained within a database became increasingly apparent as the *Lexpert* part of project CLEAR wore on. For example, as I originally envisioned *Lexpert*, it would lead users down the decision tree to an answer recommending a particular section of a particular research aid. Only then would the user be able to ask "where is that aid located in this library?"

It occurred to me much later in the project that some users might know that *Lexpert* could provide location information; these users might want to access the "where is it" part of the system directly. That is, they might already know that the research aid they needed was the *CCH Congressional Index*, but they just did not know where that aid was located in the library.

If they knew that *Lexpert* held the answer to that question, they would still not be able to get to it because of the way *Lexpert* was structured. They would have to guess which branches through the decision tree would lead them to the answer "*CCH Congressional Index*", and then retrieve the corresponding "where" file.

Once I identified a need for direct access to all the "where" files, I realized that the database included all the needed information and could write it out to a disk file in the form of an index with hypertext links to all the "where" file names. The template for such a report, associated with the publications database, was easy to create and includes specifications (1) to sort entries alphabetically; (2) to leave a space whenever the first letter of the name of a publication changes; and (3) to paginate for 24 lines per "page" because that is how many lines can fit on a single computer screen. Simplified to its essentials, a schematic illustration of the template is shown in Figure 22.

Figure 22: Database template for producing an index to all "Where" files.

<u>Name of the publication</u>	<u>Location info</u>
[PUB_NAME]	<whe[PUB_NUMBER]>

When this report is run, it produces a file that I labelled "INDWHE" for "index to the where files." A portion of that file is shown in Figure 23.

Figure 23: An excerpt from the index to all "where" files produced as a report from the template shown in Figure 22.

<u>Name of the publication</u>	<u>Location</u>
Ames's Comprehensive Index: 1881-1893...	<whe12045>
Annals of Congress	<whe12009>
CCH Congressional Index.....,	<whe12003>
CIS Index to Unpublished US House Committee Hearings	<whe14008>
CIS Index to Unpublished US Senate Committee Hearings	<whe14007>
CIS Presidential Executive Orders and Proclamations	<whe14005>
CIS U.S Serial Set Index	<whe12004>
CIS US Congressional Committee Hearings Index	<whe14006>
CIS US Congressional Committee Prints Index	<whe12015>
CIS/Annual Index to Publications of the U.S. Congress	<whe12001>
Carroliton Press Index to Monthly Catalog	<whe12021>
Checklist of U.S. Public Docs. 1789-1909	<whe12046>
Code of Federal Regulations (CFR)	<whe14002>
Congressional Globe	<whe12007>

Though not shown in this paper, I also set up a nearly identical index to the "what" files - the files giving a general description of each research aid included in the database. This took no more than a few minutes. Once both these report templates were defined, printing them out became an automatic procedure, accomplished by a number of "scripts" that can control the operation of the database.

Whenever the *Lexpert* decision tree is re-generated because of the addition of new information to the database, a script takes care of ensuring that the "where" and "what" indexes are freshly and accurately regenerated as well.

Conclusion

A combination of computer software programs and techniques can help automate the process of creating an expert system for giving advice about legislative history research. Although these techniques do not produce the kind of expert system that is capable of reasoning about law and legal precedents, they do allow the creation of a helpful system, one that embodies a substantial amount of detailed knowledge about legislative history research and is capable of delivering that knowledge to its users with speed and simplicity. The techniques for automating the creation of this type of expert system are as follows:

1. the use of a database to store and manage a list of research aid titles and the characteristics that make them appropriate for use, and to write out "reports" to disk files that become the input files to a program that creates decision trees;
2. the use of a decision tree-creating program that can read the list of research aid titles and characteristics, then convert that list to a decision tree implemented as a hypertext system of linked files;
3. the creation of the decision tree as separate text files and the ability to use ordinary word processing software to improve the format and readability of these files;
4. the use of other database reports that become disk files to provide information collateral to the decision tree itself but "hypertext linked" to it, such as indexes to the system's information; and
5. the use of a simple hypertext file-reading program as the means for delivering the expert system to researchers so that they are able to obtain its advice about legislative history research.

The techniques just described have been used to create the *Lexpert* system for advising researchers on how to find the variety of documents that make up a federal legislative history.

Equally important, these techniques also lend themselves to the development of other expert systems for giving legal research advice.

The advantages to such techniques are many. By starting with a database, an expert system developer can manage the system's information through ordinary database techniques: sorting, searching, printing out reports. New entries can be added over time and checked for consistency with the existing entries' spelling, punctuation, etc. Whenever the developer has accumulated enough entries in the database about a new set of research aids to make it worthwhile, the database can print out the appropriate files to disk using report templates designed and specified ahead of time. Each new round of reports is therefore produced automatically and with correct hypertext linking to other reports.

The files thus produced become both the core decision tree of the expert system, as well as its collateral information. Collateral information includes such things as files that recapitulate a researcher's request for advice; that offer further instruction on what a particular research aid is for, how to use it, and where it is located in the library; and that index some of the system's information. The format of these automatically-produced reports can be altered at any time, with the alterations taking effect when the system is next regenerated. That allows changes to the format and content of the expert system's files to be made with certainty that all files will acquire the new format and information and hence have a consistent appearance.

The implications of this approach to creating expert systems in general are several.

First, useful expert systems can be created in law library research that do not depend on highly complex software to deliver the system to its end users. Most conventional expert systems rely on software to draw conclusions each time an end user asks a question. The fact that the system must draw conclusions each time makes such expert systems flexible and powerful, but also means that they are computing intensive and hence require more powerful computer hardware than most hypertext systems require. The software complexity in *Lexpert's* hypertext approach shows up only in the development cycle and even there consists in the integration of a large number of individual tools, each of which is fairly simple and understandable on its own. The particular distribution software for *Lexpert* is a straightforward and reliable hypertext reader program that requires minimal computing hardware.

Second, the consistency and regularity forced into the system by the database report templates offers the possibility that the information produced might be usable with a variety of hypertext programs. In particular, it appears possible to convert the large number of individually small linked files produced by the *Lexpert* techniques to run with any of several other hypertext packages besides the one actually used for the project. This means that conversion to other hardware and software systems should be straightforward. Indeed, it may even be possible to modify the database technique to produce reports that are already in the format needed by other hypertext packages. This approach would allow the creator of an expert system to by-pass the production of the small text files altogether.

The use of a database allows a developer to separate two things: (1) the collection and organization of the expert knowledge required for legal research and the production of a decision tree embodying that knowledge; and (2) the decision about what software package will deliver the system's advice to researchers. Conversion to other packages would permit the use of expert legal research advice systems on different computing hardware or with different software environments. At the very least, this flexibility should allow the decision tree to be incorporated into delivery software that can add features such as password control or greater user interaction, hence adding greater value to libraries

and researchers.

Finally, a database is useful as a general means of managing and standardizing an expert system that depends on large amounts of text about a particular area of knowledge being entered and spelled and capitalized in a consistent way. This component of the *Lexpert* techniques would therefore be useful for the type of expert systems that use a knowledge base and separate rules of logical inference rather than hypertext as the delivery mechanism.

Insofar as these more "traditional" types of expert systems depend on stringent text management during the development process, a database much like the one described in this paper could help manage that system and thus automate the process of creating both the knowledge base and the rules of inference used by such a systems.

Footnotes

1. Associate Professor of Law, College of William & Mary, Williamsburg, VA 23185. Phone (804) 221-3826. Internet: Thardy@wmvm1.cc.edu. BitNet: Thardy@wmvm1.bitnet. Thanks to David Reed, Walter Echwald, Edward Richards, and David Burch for helpful comments on eariler drafts

2. Throughout this paper, I will mention the names of several computer products. I have no affiliation with the companies that make these products; I derive no remuneration from the mention of their products

3. One short definition of "expert system" is "a computer program that relies on knowledge and reasoning to perform a difficult task usually undertaken only by a human expert." K. Parsaye & M. Chignell. *Expert Systems for Experts* 1 (1988) (emphasis in original). See the additional discussion of facts and rules (which correspond to "knowledge" and "reasoning") contained in Id. at Chapter 2.3, *Structuring and Representing Knowledge* at 35. "Reasoning" in expert systems typically means reasoning at the time a user asks questions of the system. *Lexpert's* decision tree approach can be thought of as carrying out its "reasoning" at the time the system is created.

4 At this writing the decision tree alone consists of 179 files totalling 247,900 bytes, for an average file size 1385 bytes The *Lexpert* system overall contains 663 files totalling 850,396 bytes, for an average file size of 1283 bytes

5 For an overview of what "hypertext" is, see Hardy, *Project CLEAR's Paper Choice: A Hypertext System for Giving Advice About Legal Research*, 82 Law Library Journal 209, 210-211 (1990). For a more extended discussion, see Conklin, *Hypertext: An Introduction and Survey*, Computer, September 1987. For excellent advice on creating hypertext systems, see J. Martin, *Hyperdocuments & How to Create Them* 33 (1990). Two good collections of hypertext papers are: E. Barrett (ed.), *Text Context, and Hypertext: Writing with and for the Computer* (1988) and E. Barrett (ed.), *The Society of Text Hypertext, Hypermedia, and the Social Construction of Information* (1989).

6 See Hardy, supra note 5. See also Hardy, *A Hypertext System far Teaching Legal Research*, in S. Nagel, ed., *Law, Decision-Making. and Microcomputers* 195(1 991).

7 See Conklin, supra note 5, at 38; J. Martin, supra nate 5 at 33.

8 The problem is hinted at in Shneiderman, *Reflections on Authoring, Editing, and Managing Hypertext*, in Barrett, supra note 5 at 115,121.

9 I ws able to graft the *Lexpert* system onto the existing Paper Choice system to see if the two could

co-exist, since they are based on nearly identical hypertext technologies. Together they comprise over 900 files containing more than 3,300 individual hypertext links

10 Other characteristics like "color" or "whitewall tires" could of course be used; the only criterion for choosing which characteristics should be listed is that the characteristic should be relevant to the choice at hand. If buyers of, used cars are concerned about the length of a warranty, for example, then a "Length of warranty" characteristic should be added to the list. If buyers do not care about miles per gallon, then "MPG" should be removed from the list.

11 "Perm" is available from the MaxThink Company, 2490 Channing #218, Berkeley CA 94704. See disclaimer in footnote 2.

12 The CIS materials began coverage in 1970. Up to 1983, the relevant legislative history materials were listed in a volume labelled "ABSTRACTS." After 1983, CIS began issuing a separate volume called "LEGISLATIVE HISTORIES," so that directions for using an "ABSTRACTS" volume would no longer be applicable after that date.

13 The Perm program requires some additional information beyond what I have shown here. This additional information is described in note 23. Also, the program's actual output does not include the lines shown in the Figure.

14 The Perm program is limited in the amount of characteristics that it can handle, but I have not yet reached that limit with 128 individual pieces of advice.

15 I am currently working with about 128 pieces of discrete advice or answers, each of which has eight characteristics, though at present only four of these are used to create the resulting *Lexpert* system.

16 Please see the disclaimer in note 2.

17 Like any good organizer of information, I tried at first to make the publication numbers significant. An initial "1" in a publication number meant the work dealt with Federal materials; a "2" meant State materials; "3" meant local; etc. The second digit was designed to reflect the type of material covered: a "1" meant Constitutional authority; "2" meant statutory or legislative authority; "3" meant case law or judicial authority; etc.. In practice, my undertaking, which included local, state, and federal materials of Constitutional, legislative, and judicial scope, proved far too ambitious. I scaled back to concentrate solely on federal legislative materials. Consequently, most of the publication numbers in the publications database begin with "12" for "federal legislative" materials

18 Readers familiar with database technology will recognize this arrangement as two "tables" in a "relational database system."

19 "R" was picked as an abbreviation for "record" because I initially referred to the advice database as containing "record numbers" and the publication database as containing "publication numbers." In fact, both are "record" numbers, but the initial choice of abbreviation acquired familiarity and so continued in use.

20 See Hardy, *supra* note 5 at 216-218.

21 The terms are distinguishable by the Perm software, however, and in any event they are changed in the final stages of the generation of *Lexpert* to terms more meaningful to researchers.

22 A "template" is a description of the content of a report, written in a form prescribed by a particular database system. The system uses the template to determine where to put information from

the database onto the report.

23 To make use of this list of characteristics, the decision tree-creating program has to have a separate file of "category descriptions" that tell it what the significance is of a line like "JURI: Legislative materials". It is possible, of course, for a program like Perm to read the expressions "ANSWER:", "INPU:", "OUTP:", etc. and treat those prefixes as field or category names. That would avoid the need to create a separate file of category descriptions. Unfortunately, Perm is a commercial program that users are unable to modify and it is programmed to require a separate file. This file lists each category of information with an arbitrary label, followed by all unique entries under that category.

Like the list of characteristics file in Figure 8, this file is itself produced automatically from the database. Unlike with the other reports, the task cannot be accomplished with a single template; a separate template for each field is necessary. (At least I found it to be necessary. Readers more knowledgeable about Paradox than I could perhaps find a way to do it better.) All four reports are written to disk as plain ASCII files, however, so that it was a simple matter to create a DOS batch file that copies each of these four separate files onto a single category description file.

The Perm program needs one other file: a list of which questions should be asked first in the decision tree. This file can contain more than one sequence of questions, so that multiple decision trees can be created.

24 Outlining programs are roughly comparable in cost and nearly so in complexity to word processing programs. Requiring users to buy one to read a decision tree would be like requiring them to buy a word processing program just to read a text file.

25 See supra, note 11.

26 The macro program is written for Microsoft's "WORD for DOS" word processing program.

27 The amount of elaboration is greatest with the files that yield the "ANSWER" to the user's research question (compare Figures 13 and 17). It may seem silly to go to such lengths to emphasize what ought to be obvious: that the answer is the answer. Unfortunately, experience has shown that users of hypertext systems often settle into a state that I call "hypertext hypnosis," akin to highway hypnosis: they do not realize what they are reading even as they read it.

With earlier, simpler versions of the "ANSWER" screen, researchers sometimes kept pressing keys as though more hypertext jumps were necessary. Hence the elaborate format of the "ANSWER" files shown in the text was needed to make them realize where they were.

28 See J. Martin, supra note 5 at 97-98 (importance of good visual representations); 15-16 (selection of typefaces); 129-30 (use of color). Martin compares the need to present concise information in hypertext systems with an advertiser's need to present concise information to potential customers. He recommends that hypertext creators... study advertisements and how their authors use English, their headlines, their copy, their layout, and their graphics:" Id. at 98.

29 The decision tree is created as a pure hierarchy; there is only one way to get to a particular leaf at the end of a series of tree branches.

30 The report is generated initially as a single file that contains all "rec" reports for the entire database. The Hyplus hypertext system requires, however, that each record be contained within a separate file. Thus the actual procedure has an extra step: a small program reads this single report file and breaks it up into its constituent pieces, writing each of these to individual files: By including the record numbers in the larger report, the "splitter" program can assign file names that are keyed to the record numbers: Hence the files are titled with "rec" plus a record number.

31 Unlike dBase and its clones, Paradox does not allow "memo" fields for text storage.

32 See for example the work described in K. Ashley, *Modeling Legal Argument Reasoning with Cases and Hypotheticals* (1900). Ashley has developed techniques that comprise

- a conceptual structure for case-based legal arguments;
 - computational definitions of how cases are relevantly similar and different;
 - a computational definition for qualitatively measuring which cases are... most on point... with respect to a problem and which cases are the best to cite in favor of a party's position;
 - ...; [and]
 - knowledge representation structures and a control structure for implementing the definitions so that [a computer] program can select relevant cases, choose the most-on-point cases, cite them in arguments, distinguish them, pose cases as counterexamples, and pose hypotheticals based on precedents to show how to strengthen an argument
- Id.at4.

My work with Lexpert is less complex than this, coming closer to the "logic programming" work done by Kowalski and Sergot at the University of London. See Kowalski and Sergot, *The Use of Logical Models in Legal Problem Solving*, in A. Narayanan and M. Bennun (eds.) *Law, Computer Science, and Artificial Intelligence* 90. See especially their discussion of a "program which operates by blind, mechanical application of rules," noting that the usefulness of such programs is precisely that "in the day-to-day practice of law, there are many situations where routine tasks do have to be performed, and where rules and regulations do have to be applied mechanically." Id. at 103.

Author

I.T.Hardy,

Associate Professor of Law,

College of William & Mary,

Williamsburg,

VA23185.

Phone (804) 221-3826.

Internet: Thardy@wmvm1.cc.edu.

BitNet: Thardy@wmvm1.bitnet.