Charles Kingston,[1] D.Crim.

Expert Systems in Forensic Science

REFERENCE: Kingston, C., **"Expert Systems in Forensic Science,"** *Journal of Forensic Sciences,* JFSCA, Vol. 35, No. 6, Nov. 1990, pp. 1404–1413.

ABSTRACT: Expert systems provide a means for experienced experts to make their expertise available to others. The workings of a simple expert system shell are described using several small knowledge bases for examples. Several types of materials are described which would appear to be excellent candidates around which to develop knowledge bases to assist in the identification, evaluation, or interpretation of evidence involving such materials. The knowledge bases could be used in practice by forensic scientists or for educational purposes.

KEYWORDS: forensic science, information systems, expert systems, expert system shell, expert knowledge, training, education

During a short project by a graduate student exploring means of quantifying the similarity (or non-similarity) of gas chromatography (GC) pyrograms, the question arose as to whether an expert system would be of any value for that purpose. An ensuing exploration into the uses and capabilities of expert systems resulted in a negative answer to that question. However, this exploration did show that expert systems could be useful in other areas of forensic science. The present paper was written to introduce the reader to expert systems and their potential use in forensic science.

A simple expert system shell[2] was written to provide a working program which was used during the above-mentioned exploration. The program is called "theExpert" and was written in C for the Macintosh computer.[3]

An expert system can be thought of as a means of storing the knowledge and methods of one or more human experts in a specific area, and making these available to others for guidance in the subject area or for educational purposes. It follows that the knowledge and methodology encoded into an expert system, or into a knowledge base used by an expert system, must already be known. Expert systems are not readily adapted to the

Received for publication 13 Nov. 1989; accepted for publication 19 Dec. 1989.

[1]Professor of criminalistics, John Jay College of Criminal Justice, City University of New York, New York, NY.

[2]A shell in this context is a program which is completed by the user entering information specific to a particular use.

[3]A demonstration of theExpert will be uploaded to the GEnie Information Service by the time this paper is published; it can be downloaded by the reader for personal exploration in the operation and use of expert systems in forensic science. Some sample knowledge bases illustrating forensic science uses will accompany the program. A full version of theExpert will be made available to any forensic science laboratory wishing to explore the use of larger knowledge bases and to participate in beta testing of the program. To obtain a copy, send a request to the author on your laboratory's letterhead, along with a 3½-in. disk formatted with a Macintosh computer and a stamped and addressed mailer for return of the disk.

This author has used VP-Expert by Paperback Software on an (IBM-compatible) PC and found it to be a reasonably good commercial program at a reasonable cost.

generation of new knowledge or methods, which was the goal of the project that triggered this exploration.

Results

The most common way of representing an expert's knowledge is by formulating this knowledge into rules. Rules define the relationships between what an expert observes and his or her conclusions. The way the rules are structured and the order in which they are placed in a knowledge base reflect the method or methods used by the expert. Let us look at a simple example.

Consider the development of an expert system for assisting in a screening analysis for drugs using color tests. Assume that the test reagents are labeled TR1, TR2, TR3, and so on, and that the drugs screened for are labeled D1, D2, D3, and so on. Part of an expert's knowledge would be the colors or reactions that occur when each reagent is applied to a sample of each drug. This knowledge can be expressed by rules of the form: IF *premise* THEN *conclusion*. For example:

RULE1:
IF TR1 develops a blue color,
  AND TR2 develops a brown color,
  AND TR3 develops a yellow color,
. . .
THEN the material may contain D4.

RULE2:
IF TR1 develops a green color OR TR1 develops a brown color,
  AND TR2 develops a brown color,
  AND TR3 develops no color,
. . .
THEN the material may contain D9.

The method used by the expert can be partly reflected in the order of the premise components. In the examples of rules above, the order was simply the numbering order of the reagents. And, indeed, the expert may apply them in that order. However, the expert may have found that another sequence of application will frequently result in a conclusion before all available reagents have been applied. In that case, the premise clauses could reflect that order, which may result in more efficient guidance being given by the expert system. This would also depend upon the operation of the inference engine of the expert system (which is another way of saying that it depends on how the expert system is programed).

Rules formatted as those listed above appear similar to the "IF . . . THEN" structure in a computer language such as BASIC, and it is possible to take such rules and use them with very little modification in a computer program. But the rules themselves do not make a complete program, and considerable programing would have to be added to provide an interface to the user for gathering facts (for example, the color developed with TR1), to interpret the rules properly, and to present the results to the user.

At one time, this was the way that expert systems were built, although a language such as LISP would generally be used rather than BASIC. Since the programing required would be similar for different expert systems (except for the rules in the knowledge base), expert system shells which incorporated nearly all the programing needed for handling a knowledge base were developed. These shells can be used to develop expert systems in a variety of areas by adding different rules. They also open the way for experts in a particular area to develop an expert system in their area even if they do not know how to program a computer. Designing an efficient knowledge base, or set of rules, for an expert system shell does require some knowledge of how the shell's inference engine and user interface work.

Let us look at how a typical shell might handle the two rules listed above. In general, the rules are examined in the order in which they occur in the knowledge base. Thus the shell would try to establish whether D4 was in the material by examining RULE1. To do this, each of the conditions listed in the premise would be evaluated. These conditions are also generally examined in the order in which they occur. So the shell would evaluate TR1, find out that it has not been assigned (or bound to) a value, and would try to establish a value for it. What happens then would depend on what other rules have been defined. If TR1 is listed as a conclusion to another rule, then that rule will generally be evaluated. Otherwise, the user will be asked to supply a value for TR1.

After obtaining a value (that is, color) for TR1, a fact is established that TR1 gives that color. The next action of the shell may depend upon what that fact is. If the fact is "TR1 gives a blue color" as required by the condition, then TR2 will be evaluated, and so on. If the evaluation of a condition in the premise does not result in a fact that meets that condition, then the conclusion will most likely be considered false and the next rule will be evaluated.

However the conclusion is evaluated, the shell will generally evaluate the next rule. When evaluating RULE2, the shell will proceed to evaluate each condition in the premise. So the shell would evaluate TR1 and discover that a fact giving the color developed by TR1 has already been established. That fact would be compared with the condition, and action would be taken depending on whether or not it matched. This process would proceed until all reachable rules had been evaluated, at which time the results would be presented to the user. Some shells may present each true conclusion to the user at the time it is established as true and then proceed or ask whether or not they should proceed.

The process described above is generally known as "backward chaining." An alternative to this is generally known as "forward chaining." In a forward chaining mode, the shell would generally ask the user initially to enter the color developed by each test reagent. This would establish a list of initial facts. It would then run through the rules, trying each premise to see if it matches the fact list. The shell may or may not be set to ask about any properties or variables evaluated for which a value has not been given.

When operated in a forward chaining mode, an expert system may not appear to be much different from a database program. Even in a backward chaining mode, an expert system may seem to be much like a database (or file manager) program that asks questions. This is particularly true with a simple knowledge base such as the example described here for color tests. One difference is that a database contains a series of records that contain facts and figures, whereas a knowledge base contains small units of knowledge. The difference becomes more apparent as the knowledge base and the logic therein become more complex.

A sample knowledge base with actual color test rules with a format similar to those listed above will be included with the demonstration version of theExpert. The format for the rules used by theExpert is not the same as that of the sample rules shown above. The rules list the conclusion first, followed by the premise conditions between parentheses. Thus, the above rules might be written as follows:

D4 (TR1 ⟨blue⟩, TR2 ⟨brown⟩, TR3 ⟨yellow⟩, . . .)
D9 (TR1 ⟨green⟩|TR1 ⟨brown⟩, TR2 ⟨brown⟩, TR3 ⟨none⟩, . . .)

The comma in the premise list indicates "AND," while the vertical line indicates "OR." Properties connected with ANDs must all be true for the conclusion to be true. Only one of the properties connected with ORs has to be true to make the group of properties in the OR chain true. Conclusions (such as D4 and D9) will be referred to as objects, and the premise conditions will be referred to as properties or variable properties. The angle brackets tag a property as a variable (property). Hereafter in this paper, rules will be expressed in this form.

Color test rules like those shown above represent a simplistic approach to such tests. Color tests may develop an initial color which quickly turns to another color or shade and which may eventually turn to some final color, perhaps looking like mud. This could be incorporated into a knowledge base by defining three variables for each test reagent, for example, TR1_init, TR1_next, and TR1_final. These could be listed in the main rules, or the values for the test reagents could be number codes generated from secondary rules within which the three values would be evaluated. The latter would considerably reduce the size of the knowledge base in comparison with the former.

Let us consider another simple knowledge base which will illustrate the use of secondary rules and will provide further insight into the operation of an inference engine. Two of the sample knowledge bases provided with theExpert are called "Agglutination" and "Blood Typing (Abs-Elu)." The base "Blood Typing (Abs-Elu)" interprets the results of typing dried blood using the absorption-elution method and does not use secondary rules. One is asked to indicate which cells showed agglutination on a graphics screen. The blood type that would show such agglutination is then printed on the screen. The knowledge base was originally limited to the information given in tables in articles or books describing the method. It was then extended to consider what would happen if the dried blood stain were a combination of two different bloods, with the tentative assumption[4] that any cross agglutination between the bloods would be undone in the sample preparation stage.

The results were originally tabulated in a fairly large table. The knowledge base Agglutination was then made to check on the results. The base operates in the forward chaining mode and initially asks for the blood types that are in the stain. It then shows which cells will show agglutination with the absorption-elution method of typing. Which cell types show agglutination will depend upon which antibodies are present after elution. The rules for this are the following (capital letters = antigen, small letters = antibody):

R_A1_a "A1 cells will show agglutination with Anti-a"
  (Abs_a|Abs_a1)
R_A1_a1 "A1 cells will show agglutination with Anti-a1"
  (Abs_a1)
R_A2_a "A2 cells will show agglutination with Anti-a"
  (Abs_a, Abs_h)
R_B_b "B cells will show agglutination with Anti-b"
  (Abs_b)
R_O_h "O cells will show agglutination with Anti-h"
  (Abs_h)

The object form is "R_" for rule, then the cell type and then the antibody type if it varies. The string in quotes after the object is what is displayed on the screen if the cells will show agglutination. The antibody types that would result in agglutination of the cell type in the rule are listed as the required properties on the following line. Thus the first rule is interpreted to say that if Anti-a OR Anti-a_1 antibodies were adsorbed (and eluted),

[4]The validity of this assumption has not been tested. It may or may not be true, depending upon the conditions to which the stain has been subjected and the experimental approach used for typing.

then A_1 cells will show agglutination. The third rule says that if Anti-a AND Anti-h antibodies were adsorbed, then A_2 cells will show agglutination.

After getting the blood types in the stain, the shell will start evaluating the rules. In the first rule, it needs to know if Anti-a or Anti-a_1 antibodies were adsorbed (if Abs_a or Abs_a1 are true). It starts with "Abs_a." Since no facts have been declared to tell it, it must go elsewhere for the information. Some secondary rules are added as follows:

/Abs_a (Ag_A|Ag_A1)
/Abs_a1 (Ag_A1)
/Abs_b (Ag_B)
/Abs_h (Ag_H)

These secondary rules start with a slash, which indicates that they are secondary rules and are not to be reported. The conclusion (object) of a secondary rule is considered to be a secondary property since it will generally be used to extend the logic for a main property. The shell looks through these rules and finds /Abs_a, which matches Abs_a when the slash is removed. The rule says that Abs_a is true if the antigens A or A_1 are present in the blood (Ag_A or Ag_A1). Now the shell is working on Ag_A, and cannot find any relevant facts. This requires another set of secondary properties to provide a result:

/Ag_A (Type_A1|Type_A2|Type_A1B|Type_A2B)
/Ag_A1 (Type_A1|Type_A1B)
/Ag_B (Type_B|Type_A1B|Type_A2B)
/Ag_H (Type_A2|Type_O|Type_A2B)

The first of these secondary rules says that, if type A_1 or A_2 or A_1B or A_2B blood is present, then Ag_A is true (that is, A agglutinins are present), otherwise it is false. In general, if a secondary property is evaluated and its properties (called subproperties) do not match the facts, and there is no other place to go to check on them, then the secondary property is considered false. If one of the listed blood types is true, then the fact that Ag_A is true will be established. If none of the listed blood types is true, then, since there is nowhere else to go for the information, the fact that Ag_A is false will be established.

If Ag_A is true, then Abs_a will be established as true, and, finally, R_A1_a will be established as true and selected for output. The shell will then go on to the next rule and evaluate it.

Suppose that Ag_A is false. Then Abs_a cannot be established as true yet. Since Ag_A1 can also be used to establish Abs_a as true (they are connected by an OR), it will be evaluated. Note that if the logic relationship were an AND, Ag_A1 would not be evaluated since Ag_A would have to be true for Abs_a to be true. When the shell reaches the last set of rules, it will establish that Ag_A1 is false and proceed to evaluate the next rule. (Follow the logic through and see if you reach the same conclusion.)

The manner in which an expert system shell interfaces with the user is important. Many shells, including theExpert, provide a means of getting input for the value of a property by allowing the user to point to the appropriate value (using a mouse or tablet) in a graphics display. This would be quite useful for a knowledge base for interpreting crystal test results. The user could point to the crystals in a display that were closest to those observed. Graphics output is also of value for displaying a picture of the result rather than plain text output. The standardization of the graphics interface throughout all models of computers was one of the main reasons for choosing the Macintosh computer for the development of theExpert. Such standardization is not a characteristic of the IBM PC and the line of computers compatible with it.

Two more sample knowledge bases provided with theExpert demonstrate different ways in which spectral data can be handled with an expert system. These are the "Short IR Base" and "IR Tree" bases.

The first is a listing of drugs and the six most intense IR peaks for each. It is used in a forward chaining mode with the peaks on the unknown spectrum entered prior to the search. A tolerance value can be set to allow for variations in the values read from the spectrum. The drug or drugs with matching peaks are displayed. This offers a simplified alternative to the dedicated spectra searching programs which has the flexibility to handle a wide variety of spectral or similar types of data. IR, UV, and XRD knowledge bases have been generated for theExpert, and these will eventually find their way into use by the forensic science instrumentation class at the John Jay College of Criminal Justice. This type of data search is not normally done with expert systems, and theExpert had to be modified to allow it.

The second knowledge base, IR Tree, demonstrates a different way of organizing and searching spectral data, which is well suited for an expert system. This approach was presented in an infrared bulletin by Hannah and Pattacini [*1*]. The spectra are sorted into bins in a binary tree based upon the presence or absence of peaks within specified ranges. The first branch in the chart presented in the bulletin splits the spectra (based on the 55 compounds that they used) into a group with compounds having a major peak between 1680 and 1725 cm^{-1} and a group without a major peak in that range. Each of these groups is further divided into two groups based upon other peak ranges, and so on.

This approach would generally be used to place the unknown into a bin containing a small group of compounds. The members of this small group would then be examined further by other means to see if any matched the unknown. With sufficient branches, it would be possible to have each compound in its own bin, but that would generally not be an efficient use of the method.

An abbreviated listing of the IR Tree knowledge base is shown below, followed by a brief discussion of it to illustrate further the operation of an expert system.

```
$ (Is there a peak [cm-1] in the region)
Cocaine (A)
Reserpine (B)
Amobarbital (C)
Phenobarbital (D)
Procaine_HCl (E)
Heroin_HCl (F)
D-amphetamine (H)
Darvon (I)
/A (:1680_1725, :1600_1615)
/B (:1680_1725)
/C (:1680_1725, :3180_3250, :1750_1765)
/D (:1680_1725, :3180_3250, :1700_1730)
/E (:1680_1725, :3180_3250)
/F (:1600_1700)
/H (:695_705, :740_750)
/I (:695_705)
```

The names of the drugs are the main objects (underscores are used between words as spaces are not generally allowed in names and properties). The capital letters are properties representing the bins defined by the binary tree. The entires with the numbers (for example, :1680_1725) are the wave number ranges that define the branches of the tree. The initial colon is used so that the initial character for those subproperties is not a digit (a requirement for proper interpretation when loading the knowledge base). The string between parentheses after the $ in the first line is used as the question leader.

The knowledge base is evaluated using backward chaining. "Cocaine" is examined first, and the fact base is searched to see if the property "A" has been declared. Since it has not at the start of the query, the secondary property list is searched to see if A occurs there. Since it does, its subproperties are then evaluated. The fact base is searched to see if ":1680_1725" has been established as true or false. Such a fact will not be found at the start, so the secondary property list is again searched to see if :1680_1725 occurs as a secondary property. It does not, so the user is asked if a peak occurs in that range:

Is there a peak [cm-1] in the region :1680_1725.

There will be five choices that the user can select: Yes, No, Don't know, Why, and Cancel.

The Why choice will cause the program to inform the user as to what property of what object is being evaluated. Some expert system shells allow a paragraph or so, which offers a detailed explanation of why a particular question is being asked, to be included for each rule in the base. This helps to inform the user about the way the program is "thinking."

Assume that the answer is No. The program establishes the fact that A is false, and returns to the object and rejects it, since a required property is false. The next object, "Reserpine," will then be evaluated. The property "B" will be examined, which will lead to the secondary property list where B will be found. The subproperty for /B, which is :1680_1725, will be examined, and the fact that it is false will be discovered. Then the fact that B is false will be established, and Reserpine will be rejected, since a required property is false. This rejection of objects will continue until "Heroin_HCl" is evaluated.

Assume that ":1600_1700" is true (that is, that there is a peak in the range 1600 to 1700), and the answer to that question is Yes. The fact that :1600_1700 is true will be established. This will establish the fact that "F" is true, and Heroin_HCl will be marked as true and put in the list of results to be reported at the end of the query.

The shell will continue on to evaluate the remaining objects, but these will be rejected since :1600_1700 is true, which means that the secondary properties "/H" and "/I" cannot be true, since they do not have that subproperty. The final result is that Heroin_HCl will be reported as a possibility for the unknown.

Discussion

The above discussions should provide a preliminary understanding of the inner workings of an expert system. Some books were found to be particularly helpful for explaining different aspects of expert systems. Frenzel [2] provides a simple basic overview of expert systems. Harman et al. [3] have a chapter which catalogs many commercial applications. Of potential spin-off value to forensic science are the systems listed under the Science and Medicine Applications heading. Parsaye and Chignell [4] present an advanced and detailed discussion of the inner workings of expert systems.

References *5* and *6* are articles that relate to expert systems in general. Reference *7* discusses the general application of expert systems in law enforcement, and Refs *8* through *10* discuss various aspects of expert systems as they relate to an analytical laboratory.

The knowledge bases discussed earlier are simple examples that could be useful in a forensic science laboratory in expanded forms. Their simplicity makes them good choices as vehicles for explaining the basic ideas of expert systems. They are not, however, examples that show the full capabilities of expert systems. The information used for the knowledge bases is readily obtained from published tables, and these tables could be used to reach the same conclusions, although with somewhat more effort and difficulty.

The example knowledge bases discussed above have a straightforward logic which is much the same as that found in a database program containing the same data. With such knowledge bases, an expert system is similar to a database program that asks questions. However, expert systems are capable of handling quite complex logical interactions and relationships that would seriously tax a database program. They will generally allow one to assign a confidence factor (CF)[5] to a fact or conclusion. These CFs are evaluated during a query and provide an indication of the confidence that can be placed in a conclusion based upon the entered facts. The information stored in a database is static and has no influence on the way the database program extracts the requested data. The knowledge in an expert system is dynamic and can be thought of as a program which is run within an expert system shell. The way in which the knowledge is structured can influence the way the expert system performs its evaluation.

The example bases described above do not provide expert guidance other than that reflected in the organization of the objects and properties within the bases. The knowledge that a human expert derives from experience (sometimes called heuristics or rules of thumb) can be incorporated into a knowledge base and made available to guide the user of the expert system.

Some of the example knowledge bases could be expanded to reflect interacting conditions and methods that would make getting the same guidance from a database extremely difficult. For instance, the blood-typing knowledge base could be expanded to include information about the conditions to which the bloodstain had been subjected, the possible effects of different conditions on the results, the results of control tests, the experimental conditions used for the typing process, and the intensity of the reaction. The base could then alert the user to irregular results, offer diagnostic information as to why such results may have occurred, and so on. Conclusions suggested by the knowledge base might have varying CFs, depending upon the declared conditions. It is perhaps stating the obvious to say that the final conclusions must rely upon the human expert's own judgment and knowlege. An expert system may or may not always provide the proper answer, depending upon the type of knowledge and the structure thereof. An expert system is a tool, much as a spectrophotometer is a tool, which can be used by the forensic scientist as a guide to the formulation of a conclusion or opinion or for educational purposes. The interpretation and use of results with respect to the evaluation of physical evidence from an expert system should only be done by one who would otherwise be capable of evaluating the evidence without the expert system. An expert system which is designed for the evaluation of evidential materials is not a substitute for an expert; it is an aid to be used by an expert.

In addition to expanded versions of the above bases, other candidate knowledge bases of interest in forensic science which could utilize the above capabilities might be those that could provide the following assistance:

(*a*) help interpret blood spatter patterns,
(*b*) help determine the cause/origin of building fires,
(*c*) help analyze a general unknown (chemical),

[5]A confidence factor (CF) represents the certainty that can be associated with a fact or conclusion. A CF will have a value of 0 to 1 (or 0 to 100), where 0 represents no confidence and 1 represents the strongest confidence. CFs for facts and conclusions are assigned when the knowledge base is created (or modified).

(*d*) help determine the species of an animal hair,
(*e*) help determine the generic type of synthetic fiber (and brand?),
(*f*) help determine the species of a piece of wood, and
(*g*) help diagnose/repair instruments.

Some initial exploration was done for the development of an expert system knowledge base for helping interpret what happened at a crime scene to create the pattern of any observed blood spatters. An early suggestion was that the expert system should calculate the angles of impact and suggest trajectories and possible origins of individual blood spots (after entering in the measurements of the spots). This is the sort of task that an expert system should not be expected to perform (although for ease of use the task could be integrated with an expert system). That task can be accomplished by applying one or more formulas to the measurements. Such a task is better done with a special program, perhaps one like that presented by Wilson and Schussler [*11*]. What does seem to be needed to make that process more palatable is a program which would allow the measurements to be made using photographs of the surfaces on which the spatters occur on a graphics input tablet. Those data could then be fed directly into a program component that calculated the trajectories and possible origin areas.

This trajectory and origin data could then be fed into the expert system as a set of pre-declared facts. The expert system would then ask for whatever other information was needed to help guide the user in evaluating what happened, and in what sequence, to create the spatter pattern. As noted above, the two programs could be combined into one for ease of use.

Item 7 above (diagnose/repair instruments) represents a practical use for forensic scientists who use instruments to assist in the evaluation of evidential materials. Getting an instrument repair technician to a laboratory to repair a malfunctioning instrument may not always be expedient. Expert systems have proven to be quite useful for equipment maintenance.[6] This is an issue that should be discussed with the manufacturers of instruments used by forensic laboratories. They are the ones that have the expertise to put together a knowledge base on the instruments they manufacture.

Summary

Expert systems provide a means for experienced experts in specific areas of forensic science to make much of their expertise available to others in forensic science. The expertise is generally formulated into a series of "IF . . . THEN . . ." rules or their equivalent. With some initial training in the use of expert systems and in the generation of knowledge bases, an expert with determination can personally develop a knowledge base (or bases) in her or his particular area or areas of expertise. A number of potential evidential materials appear to be ideally suited for the development of knowledge bases for their identification and evaluation. Such bases would provide useful assistance to others in forensic science and could serve as excellent teaching tools in laboratories and forensic science classes.

References

[*1*] Hannah, R. W. and Pattacini, S. C., "Drug Identification from Their Infrared Spectra," *Infrared Bulletin No. 16*, Perkin-Elmer Corp., Norwalk, CT, April 1972.
[*2*] Frenzel, L. E., Jr., *Understanding Expert Systems*, Howard W. Sams & Company, Indianapolis, IN, 1987.
[*3*] Harman, P., Maus. R., and Morrissey, W., *Expert Systems Tools and Applications*, Wiley, New York, 1988.

[6]For example, see Ref *3*, Chapter 15, under "Equipment Maintenance Applications."

[4] Parsaye, K. and Chignell, M., *Expert Systems for Experts,* Wiley, New York, 1988.
[5] Meyer, M. H. and Curley, K. F., "Expert System Success Models," *Datamation,* 1 Sept. 1989, pp. 35–38.
[6] Casey, J., "Picking the Right Expert System Application," *AI Expert,* Sept. 1989, pp. 44–47.
[7] Reboussin, R. and Cameron, J., "Expert Systems for Law Enforcement," *FBI Law Enforcement Bulletin,* Aug. 1989, pp. 13–16.
[8] Wythoff, B. J. and Tomellini, S. A., "Expert Systems Lend a Hand in the Laboratory," *Research and Development,* April 1989, pp. 52–56.
[9] de Monchy, A. R., Forster, A. R., Arretteig, J. R., Lan, L., and Deming, S. N., "Expert Systems for the Analytical Laboratory," *Analytical Chemistry,* Vol. 60, pp. 1355A–1361A.
[10] Harrington, P. de B., Street, T. E., Voorhees, K. J., di Brozolo, F. R., and Odom, R. W., "Rule-Building Expert System for Classification of Mass Spectra," *Analytical Chemistry,* Vol. 61, pp. 715–719.
[11] Wilson, F. E. and Schuessler, D. R., "Geometric Bloodstain Pattern Interpretation Using a Computer Program," *Crime Laboratory Digest,* Vol. 14, No. 3, July 1987, pp. 95–111.

Address requests for reprints or additional information to
Prof. Charles Kingston
Science Department
John Jay College of Criminal Justice
City University of New York
445 W. 59th Street
New York, NY 10019