Journal of Chromatography, 485 (1989) 283–296
Elsevier Science Publishers B.V., Amsterdam — Printed in The Netherlands

CHROM. 21 945

EXPERT SYSTEM FOR METHOD VALIDATION IN CHROMATOGRAPHY

M. MULHOLLAND*, N. DUNAND and A. CLELAND
Philips Scientific, Cambridge (U.K.)
and
J. A. VAN LEEUWEN and B. G. M. VANDEGINSTE[a]
Catholic University of Nijmegen, Nijmegen (The Netherlands)

SUMMARY

Validation of chromatographic methods is essential to ensure that a newly developed or implemented method is capable of performing the analysis within the required accuracy and precision. A validation programme consists of the evaluation of precision, accuracy, specificity, sensitivity and limitations such as lifetimes of reagents and detection limits. These performance characteristics are affected by many contributions, *e.g.*, precision is affected by repeatability and reproducibility. The validation experiments must be designed such that these contributions are tested under the conditions with which the method is likely to be used.

The first results of research into the application of expert system technology for method validation are presented. The acquisition and representation of the knowledge are described. The expert system is designed to set up and interpret results from various method validation experiments. The expert system development tool used for the creation of a prototype is Goldworks.

INTRODUCTION

An expert system embodies in a computer the knowledge-based component of an expert's skill, in such a form that the system can offer intelligent advice and, on demand, justify its own line of reasoning[1]. The essential idea of an expert system is to take the knowledge from a recognized expert and put it into a computer system so that it can then be consulted by many other users. In theory, this has two major advantages: the expertise is much more accessible, and the experts can be freed from much of their consultancy work, thus allowing them to pursue other tasks such as research. Cleland and Mulholland[1] described the principles of building expert systems with examples in chromatography.

The major difference between an expert system and conventional software is its ability to manipulate objects and heuristic knowledge in addition to numbers and

[a] Present address: Unilever Research, Vlaardingen, The Netherlands.

0021-9673/89/$03.50

algorithms. This must be considered when selecting a suitable application. Expert systems are not the most suitable tools for solving problems that are largely algorithmic.

Several workers have published research on expert systems in analytical chemistry. Buydens *et al.*[2] described an expert system for the development of UV spectrophotometric methods for pharmaceuticals. High-performance liquid chromatography (HPLC) has been the focus of many papers[3–5], probably owing to the heuristic nature of HPLC method development. In all of these papers early expert system technology such as Prolog or KES was used[3]. Some workers tackled very carefully defined problems, *e.g.*, the planning of separations of steroids by HPLC[4]. Others have attempted to solve much larger problems but in less detail, *e.g.*, an expert system for the entire area of HPLC method development[5].

The knowledge domain described in this paper concerns the validation of liquid chromatographic methods for pharmaceutical analysis. Validation of chromatographic methods is essential to ensure that a newly developed or implemented method is capable of performing its analysis within the required accuracy and precision. Validation experiments require careful planning to test the performance of a method under the conditions with which it is to be used. Table I shows some typical validation procedures for a number of different performance characteristics. The tests required are selected by considering features of the application, *e.g.*, its intended usage and its purpose. The instrumentation available can also affect the tests which are possible[6].

The validation of analytical methods is increasingly pertinent, particularly in the pharmaceutical industry. This is due to both the potential toxicity of analytes and the controls of regulatory bodies. Several workers have recently published research in this area[7–9] which offers general guidelines for designing method validation experiments.

Expert system development software, defined as tools or shells, have recently improved markedly. There are now tools available that are more competent in dealing with mathematics and statistics, which was a significant limitation in early tools. The tool employed for this aplication is Goldworks[10], which supports the use of most knowledge representation techniques; these are more fully explained later. It can also interact with a spreadsheet package to perform mathematics and statistics.

TABLE I

METHOD VALIDATION PROCEDURES FOR DIFFERENT PERFORMANCE CHARACTERISTICS OF CHROMATOGRAPHIC METHODS

Performance characteristic	*Validation procedures*
Specificity	Interferences (*e.g.*, degradation products, process impurities); peak integrity (signal ratios with dual-channel detection, spectral comparison with multi-channel detection)
Precision	Repeatability; reproducibility; ruggedness
Accuracy	Linearity; recovery
Sensitivity	Slope of linear calibration graph
Limitations	Limits of detection and quantification; stability of samples, solvents and reagents; sample capacity; ruggedness

This combination is extremely powerful in providing solutions to the problems of method validation, which require both heuristic and algorithmic knowledge. Our early work on this combination for the validation of precision demonstrated its potential to be very encouraging[11,12]. In the work described in this paper we attempted to tackle more fully the general problems of method validation with this contemporary expert system software.

EXPERIMENTAL

Software

Goldworks is available from Gold Hill Computers (Cambridge, MA, U.S.A.) and Lotus 1–2–3, 1986 from Lotus Development (Cambridge, MA, U.S.A).

Instrumentation

A PU4100 liquid chromatograph fitted with a PU4700 autojector (Philips Scientific, Cambridge, U.K.) and an IBM/PC/AT computer with a base memory of 512K and an extended memory of 8MB were used.

BUILDING THE EXPERT SYSTEM

Defining the specification of the expert system

Fig. 1 shows the stages in the development of an expert system and their relative time scales. The first steps in selecting a knowledge domain (*i.e.*, the definition of the problem to be solved) and an expert are the most important. The domain must not be

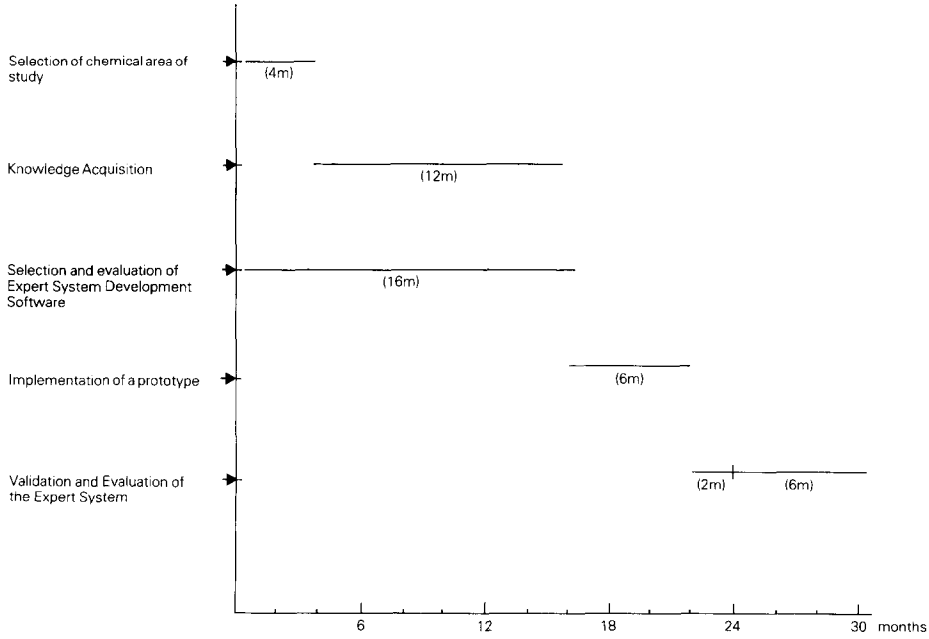

Fig. 1. Relative time scales of the stages in the development of an expert system.

too broad, attempting to cover a large problem that can only be solved in little depth; nor must it be too narrow and specific. Although it would be possible to analyse such a problem in great detail, the expert system would have few users. The application domain defined for this expert system is the validation of liquid chromatographic methods for pharmaceutical compounds. This domain can be further divided into five separate domains (specificity, precision, accuracy, sensitivity and method limitations). The advantages of doing this become clear when developing a structure for the expert system, as explained later. The overall objectives of the expert system are as follows:

(1) To give advice on the level of validation required, *e.g.*, which method limitations need to be tested.

(2)To advise on test procedures and the experimental design for specific validation requirements. For instance, diode array detection can be used to test the spectral purity of chromatographic peaks. This is a specificity test procedure.

(3) To perform any necessary calculations, such as analysis of variance for a repeatability study. This is a precision test procedure.

(4) To interpret the results and diagnose potential problems.

Acquiring the knowledge

The acquisition of knowledge for expert systems is still a novel area. Although there are some guidelines available[13], there is little background as to which methods are the most effective. For this work, the expert was a novice to the field of expert systems and the knowledge engineer, although a chemist, was unfamiliar with the expertise to be acquired. This situation is typical of many expert system projects. For the knowledge acquisition process to succeed, there seemed to be only two options: either the knowledge engineer should learn all the expertise or the expert should learn how to engineer the knowledge. In this instance, an intermediate approach was developed in which a knowledge representation process was designed in cooperation with the expert. This allowed the expert to learn basic knowledge representation techniques fairly quickly, thus being able to transfer new knowledge in an already fairly structured way. On the other hand, the knowledge engineer studied the basics of the expertise, thus being able to pose relevant and intelligent questions. This also gave the engineer a better understanding of the structure of the knowledge and thus be able quickly to identify its subdomains.

This intermediate approach worked very well for the development of this particular knowledge base. From experience outside this research, it is known that this way of working is not always feasible, particularly if the knowledge engineer has no background in the expertise.

The knowledge acquisition process took place in interview sessions between the knowledge engineer and the expert, usually with a third or even fourth person present. The first series of talks was aimed at obtaining an idea of what comprised the knowledge domain. In this stage the subdomains were defined.

The expert used these talks to become familiar with expert systems and tools for building them. It was agreed to build the system in stages, dealing with the specific subdomains.

The subdomain of precision was highlighted and two separate areas of testing were identified, the first being repeatability testing with some knowledge of reproduc-

ibility and the second being ruggedness testing[14]. This subdomain was tackled in detail together with a broad overview of method validation.

Formal representation of the knowledge base

The present authors work on part of an international project sponsored by the European Commission to investigate the application of expert systems to chemical analysis[15]. Early in the project the team identified expert and knowledge engineering pairs, of which the authors constitute one. Each pair was to acquire the knowledge for a different domain[16]. Knowledge acquired by each centre had to be available to every partner in the project. To prevent any duplication of work, it was important to have a standard method of knowledge representation. This standard should be as close as possible to an implementation in an expert system. It should also be clear about the structure of the knowledge and define which decisions were made and at which point. This is necessary to monitor correctly the processes going on in the knowledge base. As the domains, each an area of method development for HPLC, differed considerably in their type of knowledge, it is not possible to define a standard way of representing specific heuristic knowledge. However, for the description of the control knowledge, or the structure of the knowledge, there appeared to be a technique that was suitable for all domains.

Each domain consists of a number of processes that must be consulted in a certain order and a choice whether to trigger a process must be made out of a series of possibilities. This control knowledge is described by the use of a standard software development technique, which uses data flow diagrams (DFSs) and state transition diagrams (STDs).

As the use of STDs and DFDs is a well known technique in software engineering, only a brief description will be given here[17]. These two types of diagrams can be used separately but are especially powerful if they are used as pairs. This means that for every DFD there is an STD on the same subject.

DFDs are used to represent the flow of data in a process. Basically they include a process, a list of data going into the process and a list of data coming out of the process. The process can then be split up into subprocesses. Each subprocess can again be graphically represented in a DFD. The input variable for the top DFD must always be found back somewhere in one of the lower DFDs. Of course, intermediate variables will emerge in the lower DFDs, from breaking down the process into smaller subprocesses. By grouping the input and output variables by their nature or by their origin, a natural representation of the underlying links between them is reached.

A fruitful extension of this concept of DFD is the addition of a blackboard architecture to it[18]. In a blackboard, knowledge sources write their results in a common framework containing facts, and from these results the system decides which action should be taken and activates the knowledge sources accordingly. Hence the system acts as a kind of discussion leader, who reads and writes on a blackboard depending on the information it obtains from the experts. Linking this concept to the DFDs allows the input and output variables to be written to the blackboard.

The DFD alone shows nothing of the actual processes going on in the knowledge base. The points where important decisions are made are hidden in the representation. It is for the purpose of illustrating these decisions that the STDs are used.

STDs picture every possible state of the system at a certain moment and specify

the conditions that make changes from one state to another. As with DFDs there is a top process (usually called something like "solve problem"), which can be broken down into several sub-processes. The breakdown of the process into STDs should be guided by those in the DFDs. For every state it must be indicated which event caused the system to enter it and, if possible, the value of the variable to which this change is related. The actual structure of the method validation expert system is described under Results and Discussion using these diagrams.

When the structuring of the knowledge was complete, the problem remained of how to represent the actual heuristic knowledge without following the representation dictated by particular expert system development software, tools or shells. This was done using standardized rules and frames and avoiding as much as possible any notation that could not be implemented in a straightforward way with any shell. It was evident that this was not always possible owing to the multitude of different software available. The frames that were used were the simple object/attribute/value type. The simplest way to explain a frame is by usaing a specific example. A frame to represent a car would have the car as the object. The object can be defined by several attributes, *e.g.*, car manufacturer, type and colour. The values of these attributes would define a specific car, *e.g.*, a Rolls Royce, Silver Shadow, grey. These frames are very useful in describing instrumentation and even processes. More complex frames were not allowed, *e.g.*, factors to attributes, because not every shell supports this. The frames defined in this way could easily be integrated with the DFDs.

The rules were also highly standardized. It was necessary to allow the use of local variables in the rules, otherwise the required documentation would become enormous. This was not seen as a problem as all the tools considered useful for this type of knowledge support the use of variables in rules. The rules were used to determine the actual processes going on in the DFDs. Rules can also be seen as the background of the decisions described in the STD. Examples of rules and frames for the method validation system are given under Results and discussion.

The prototype

Early in the project the advantage of building a prototype became evident. It allows the expert quickly to become familiar with a real expert system. It also helps the knowledge acquisition as any misunderstandings or gaps in the knowledge are immediately apparent. The prototype was implemented in Goldworks and could then be evaluated and if necessary re-implemented.

The final stage of developing an expert system is the validation. This research has achieved a working prototype, but full evaluation and validation is still necessary.

RESULTS AND DISCUSSION

Knowledge acquisition

The knowledge acquisition process explained above was very successful for this expert system. Important factors for success were the frequent contact between the expert and the knowledge engineer, an understanding by both expert and engineer of each other's work and the use of prototypes that gave those involved an idea of a real expert system. In the talks between the engineer and the expert, it proved fruitful to have a third person present who had a certain amount of knowledge on the domain subject and on expert systems to prevent unnecessary misunderstandings.

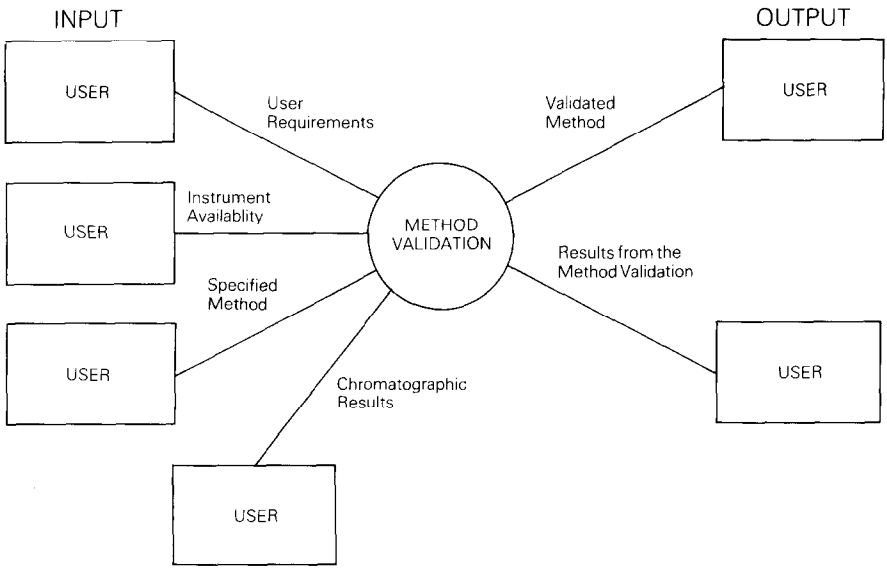

Fig. 2. Top-level DFD for the method validation system.

During the development of the repeatability knowledge base, a technique for the acquisition emerged. The expert and the engineer drew sketches of DFDs and STDs when discussing papers that the expert had prepared. From the results of these talks, real DFDs and STDs were drawn by the engineer. In the meantime, the expert could prepare new papers on the processes defined in the initial sketches. These papers were then translated into frames and rules by the knowledge engineer. Ultimately the complete description of the knowledge base resulted from this process.

The knowledge representation

Fig. 2 shows the top-level DFD for the method validation system. It illustrates the input variables coming from the user, such as usage requirements and instrument availability going into the process of method validation. The output variables emerge from this process and back to the user. The output consists of a validated method and the results from the method validation. The next-level DFD is illustrated in Fig. 3. The input variables are those of the top-level DFD. However, the sub processes now have more specific intermediate inputs and outputs. For instance, the sub process of the accuracy test has inputs from the user, but the method conditions that are used also come from the specificity subprocess. This means that any changes in the method that were necessary to achieve sufficient specificity are now implemented. This DFD can be used to illustrate how the selection of the five subdomains could be used with a blackboard architecture. Each process reads and writes information to the blackboard while it remains essentially an autonomous process. Fig. 4 shows the equivalent STD to match the DFD in Fig. 3. It clearly demonstrates the decisions made in moving from one process to another. Each test is carried out when required and only if the results are acceptable can the next test be performed.

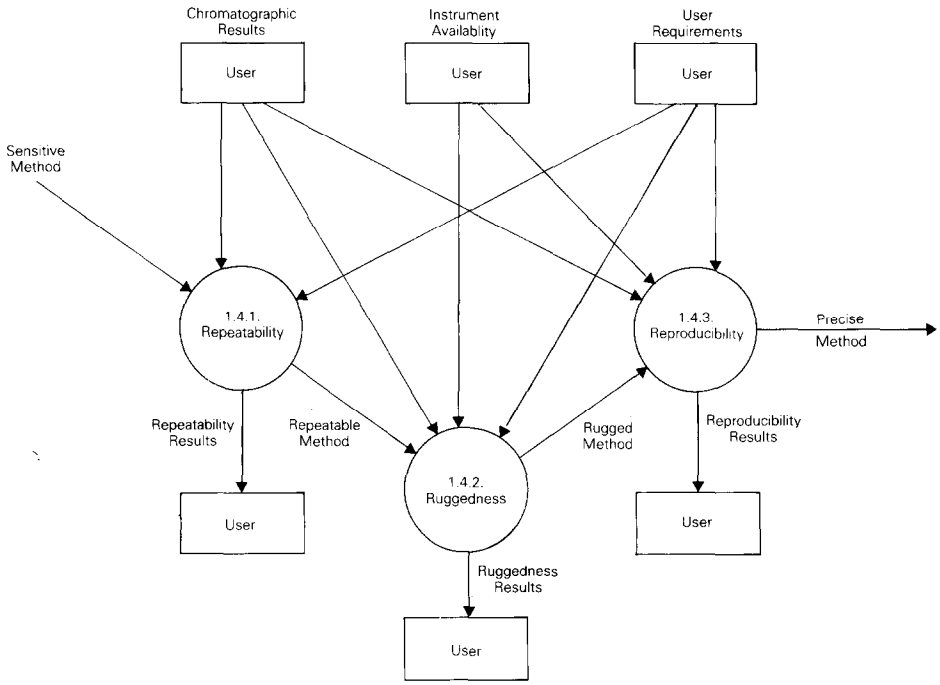

Fig. 3. DFD showing the processes of method validation.

Frames were used to represent each test and rules were built to decide which tests to perform. A method validation test became the object of the frame. Each test has several attributes and the values given to the attributes define the test. Two example tests are shown in Table II. In this way a library of tests was built up simply by defining individual values for these attributes.

Many method validation tests require the use of statistics. The Goldworks expert system software enables spreadsheets to be called up and programmed for a given test. It can then pull out results that are collected in the spreadsheet and interpret them. This was employed throughout the creation of a prototype expert system for method validation and was described in a previous paper[11].

Finally, rules were created to advise on the level of method validation required (*i.e.*, which tests are required) and to interpret the results from a specificity test. An example of such a rule is the following:

Usage of application IS long-term/inter-laboratory.
Length of run of application IS 50 samples.
Purpose of application IS NOT stability indication.
Expected interference of sample IS intermediate.
Global performance characteristics of interest of method validation system IS specificity.
CONCLUDE contributary characteristics of interest of method validation systems IS specificity.
CONCLUDE contributary characteristics of specificity performance characteristics IS interference.
CONCLUDE test procedure of interference specificity test IS blank sample preparation.
CONCLUDE test procedure of interference specificity test IS placebo sample preparation.
CONCLUDE contributary characteristics of specificity performance characteristics IS peak purity.
CONCLUDE chromatogram of peak purity test IS standard.
CONCLUDE chromatogram of peak purity test IS sample.

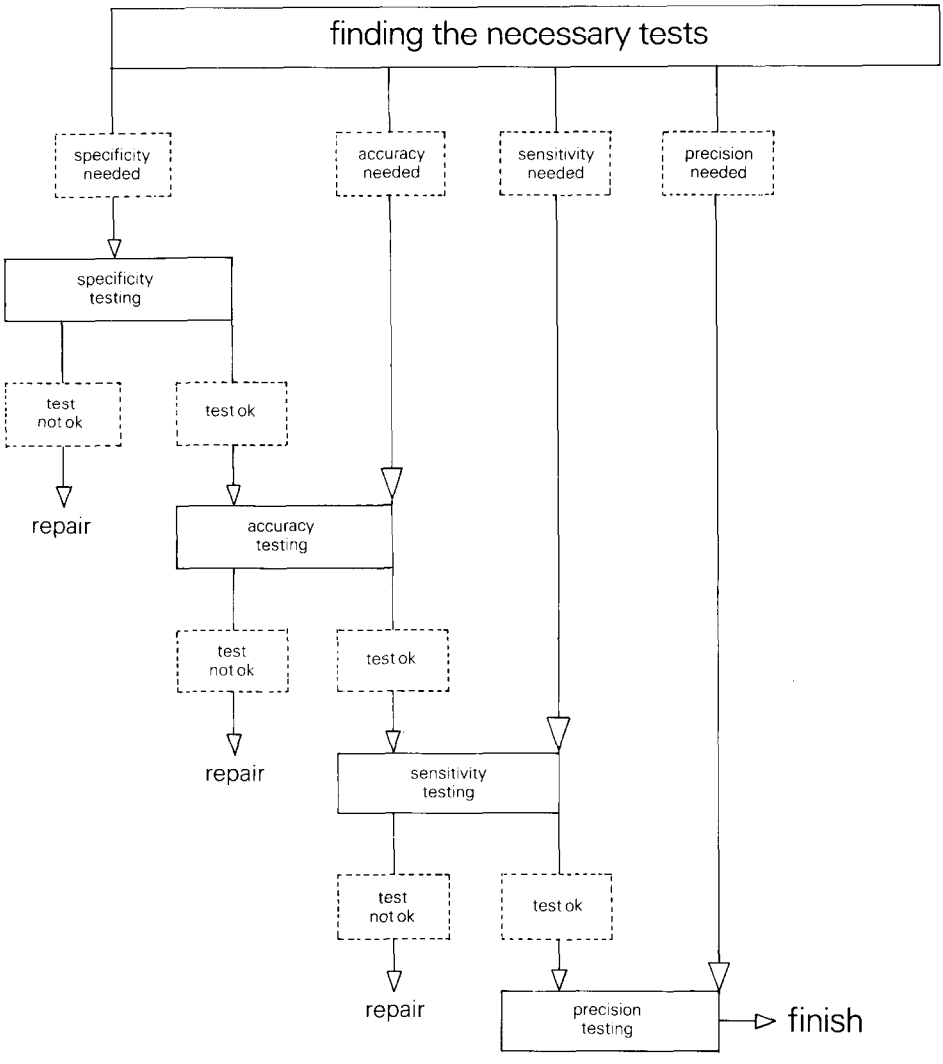

Fig. 4. An STD showing the decisions made in the selection of necessary validation tests.

This rule states that if the above conditions for usage of the method application, length of run, purpose of the application and expected interference apply, and the global performance characteristic of interest is specificity, then perform the following test: check for any interference by comparing chromatograms of both sample and standard solutions with those for blank and placebo solutions.

There was a need for rules concerning two stages in the interpretation of results. The first set of rules compare results with tolerance levels pre-specified by either the user or the expert system.

The second stage attempts to interpret the cause of a bad result and conclude a diagnostic. There can, of course, be more than one possible cause of a given problem.

TABLE II

ATTRIBUTES AND VALUES FOR THE METHOD VALIDATION TESTS

Attribute	*Values for a specificity test*	*Values for a precision test*
Global performance characteristic	Specificity; defined as the ability to determine unambiguously the component of interest in the presence of likely contaminants	Precision, defined as the random variation of the method results. It is usually a combination of the influence of all the factors in the method, *e.g.*, precision of flow-rate, injection volume, etc.
Contributory performance characteristic	Peak purity from a formulated sample	Repeatability of injection volume
Test procedure	Diode-array detection	Analysis of variance on 25 repetitions
Method variable	No value relevant	Injection volume
Chromatographic variable to measure	UV spectra	Peak areas, peak heights and retention times

If peak retention times and heights show lack of repeatability but peak areas are within tolerance, then there is possibly a problem with imprecision in solvent mixing. The following diagnostic rules would apply:

(1). Cause of imprecision of solvent mix IS air bubbles.
CONCLUDE remedy of air bubbles IS check for adequate degassing.
(2). Cause of imprecision of solvent mix IS particle impurities in pump head.
CONCLUDE remedy of particle impurities in pump head IS clean system.
CONCLUDE remedy of particle impurities in pump head IS check that HPLC-grade solvents are used.

The resulting knowledge base was successfully represented such that it could (a) be implemented into a prototype using Goldworks and (b) be transferred to all other partners in the project.

The prototype

A prototype expert system was designed with knowledge for the overall control of the five subdomains and in-depth knowledge on repeatability and ruggedness for the precision subdomain.

Fig. 5 shows an actual implementation of a frame in Goldworks. The frame shows as its object (Parent frame) the method description. The attributes were implemented as Slots, *e.g.*, flow-rate or temperature. Instances or values were selected using the instances box. For this case there was an instance called CHROM 1.

Fig. 6 illustrates an example of the implementation of a rule in Goldworks. It describes the tests to be performed when the method is required for inter-laboratory usage. The precision needed a repeatability, a reproducibility and a ruggedness study. In this way, the knowledge base of the expert system was implemented into rules and frames and with the use of object-oriented programming.

The chemical knowledge was organized around the user interface and method frames. The user interface frames controlled the input of the user and ensured that

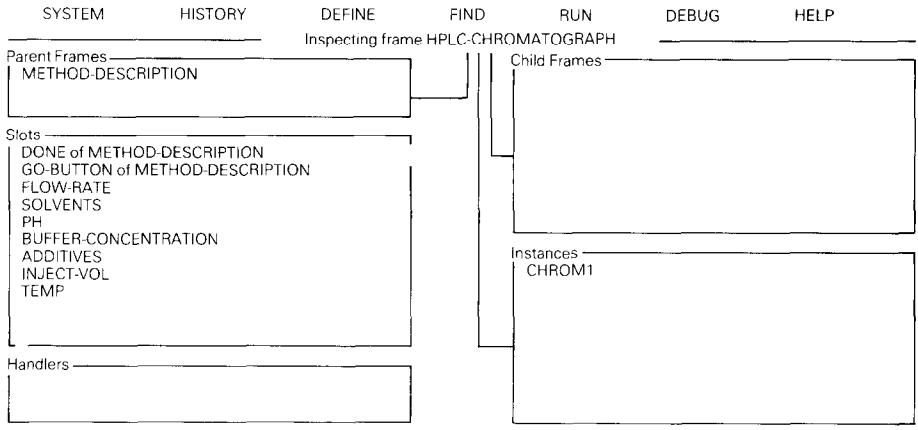

Fig. 5. Implementation of a frame in Goldworks, describing the HPLC chromatographic conditions. © 1987 Gold Hill, Inc.

only valid information was entered. Fig. 7 shows two screens of the user interface. The first page shows how the information on the method is entered; in the example, the user was in the process of entering the percentage of each solvent. The second page shows the completed input for all information concerning the chromatograph. With this information the necessary rules were triggered.

The user interface frames controlled the use of spreadsheets when required, for instance to perform calculations on the analysis of variance for a repeatability study. These frames also controlled the explanation facility of the expert system. Conventional expert system explanation facilities tend to give the user a translation of the rules that have been used in the reasoning process. This was not appropriate for this application because the user was allowed to change his answers during some of the

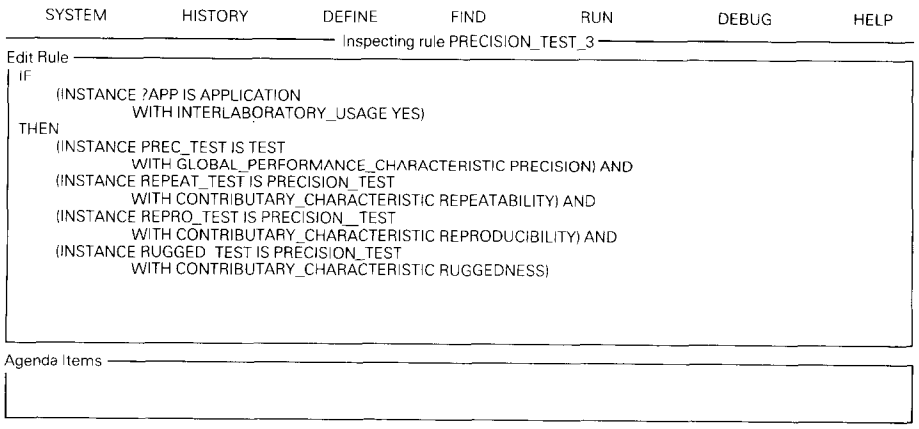

Fig. 6. Example implementation of a rule in Goldworks. This describes the necessary tests when the method is required for inter-laboratory usage. © 1987 Gold Hill, Inc.

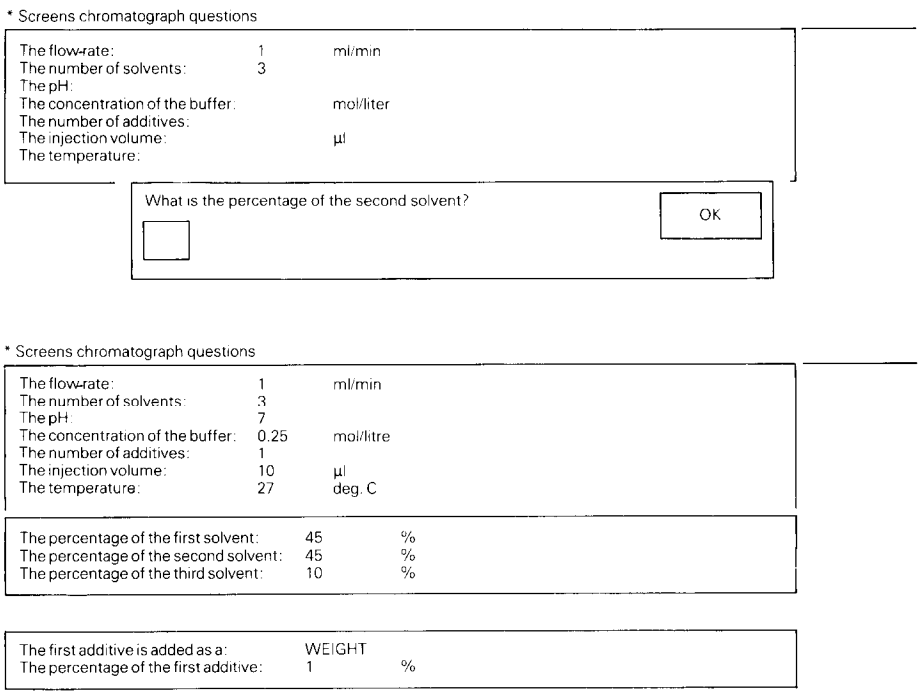

Fig. 7. Two screens showing how information on the chromatograph is input.

stages in the consultation. An explanation facility based on the rules would confront the user with a reasoning process that tried to cope with his changing insights. This system used explanations associated with conclusions. Whenever the system reached a conclusion on test set-up or diagnosis, a predefined explanation screen for this specific set-up or diagnosis is added to the list of screens that the user can select.

In order to maximize flexibility, part of the user interface was designed to finish a consultation by presenting all the input information on a worksheet. The user could change any item of information, without having a complete new consultation, and observe the new conclusions. For instance, if the user developed a need for the method to be performed in more than one laboratory, the input for inter-laboratory usage could be changed to "yes" and the new method validation requirements would be presented.

This prototype has been validated and is now under evaluation at several industrial sites in Europe. The validation of the repeatability stage was performed using the stability-indicating method for aspirin. This involved a test of ten sample injections and ten sample preparations; the results were presented elsewhere[11]. The ruggedness stage in the expert system was validated by inputting ten methods that were selected to give a wide range of pharmaceutical applications. The outputs were compared with those of the expert. The calculations were tested using three practical experiments. The first method tested was the stability-indicating assay of aspirin which involved two designs, one testing six factors and the other testing ten factors. The first design

involved duplicate experiments and fifteen data points and the second design employed duplicate experiments and fifteen data points and the second duplicate experiments at 31 data points. The second method tested employed duplicate experiments and fifteen data points and tested six factors for the stability-indicating method for salbutamol. These practical results will be used to test the diagnosis module when it is complete.

The results of these preliminary validation tests were successful. However, the evaluation of the system by independent analysts is needed in order to ascertain the real usefulness of such a system.

The prototype demonstrated the capabilities of Goldworks. Very few limitations were experienced with this application. The ability of support frames proved very effective in the simplification of the expert system. Without frames many more rules would be required. Early expert systems were extremely limited in their user interface, but this was not so with this tool. Colourful screens were programmed which allowed pop-up menus to simplify the input of information.

The flexibility introduced by the possibility of worksheet screens proved invaluable. Perhaps the most important attribute of this generation of tools is the ability to interact with spreadsheets. For an application such as method validation where statistical algorithms are required, there is now no need to write additional complex software to perform calculations.

CONCLUSIONS

The preliminary results presented here on the application of expert systems to method validation are encouraging. The knowledge and reasoning for method validation are acquired and represented successfully using the techniques described. The use of DFD and STD combinations proved particularly powerful for knowledge engineering.

The knowledge base is implemented into Goldworks with relative simplicity to create a prototype. The initial evaluation of the prototype showed it to be capable of input reasoning in a similar fashion to the expert, and making similar conclusions. The prototype can communicate with the user via state-of-the-art user interfaces which are interactive and which allow pop-up menus to facilitate inputs.

The subject domain of method validation provides varied expertise with which to evaluate expert systems. Much of the knowledge is heuristic in nature and specific to experience with particular applications. It is very difficult to use conventional software techniques to solve these problems. The ability to use spreadsheets to program the algorithms required for many method validation tests conserves considerable software effort.

The next stage of this work is to validate fully the prototype expert system. The expert must be sure, beyond reasonable doubt, that the system performs in the expected way. Following this, the prototype will require extensive evaluation of its performance in practice.

ACKNOWLEDGEMENT

This research is partically funded by the European Commission, as project P1570, Expert Systems for Chemical Analysis.

REFERENCES

1 A. Cleland and M. Mulholland, *Int. Lab.*, September (1989).
2 L. Buydens, M. Detaevernier, D. Tombeur and D. L. Massart, *Chemometrics Intell. Lab. Systems,* 1 (1986) 98–108.
3 M. A. Tischler and E. A. Fox, *Comput. Chem.*, 11 (1987) 235–240.
4 H. Garisingham, B. Srinivasan and A. L. Anandd, *Anal. Chim. Acta,* 182 (1986) 193–202.
5 R. Bach, J. Karnicky and S. Abbott, *Artif. Intell. Appl. Chem.*, 22 (1986) 278–206.
6 P. J. Schoenmakers and M. Mulholland, *Chromatographia,* 25 (1988) 737–748.
7 S. M. Ficarro and K. A. Shah, *Pharm. Manuf.* 1, Part 7 (1984) 25–27.
8 J. E. Maky and E. A. Domonkos, *J. Chromatogr. Sci.*, 23 (1985) 364–369.
9 E. L. Inman, J. K. Frischman, P. J. Jiminez, G. D. Winkel, M. L. Persinger and B. S. Rutherford, *J. Chromatogr. Sci.*, 25 (1987) 252–256.
10 P. Manchester, *P C Mag.*, 4, Part 7 (1987) 96–99.
11 M. Mulholland, J. A. van Leeuwen and B. Vandeginste, *Anal. Chim. Acta,* 223 (1989) 183–192.
12 M. Mulholland, J. A. van Leeuwen and B. Vandeginste, *Chromatographia,* submitted for publication.
13 P. Harmon and D. King, *Artificial Intelligence in Business, Expert Systems,* Wiley, New York, 1985.
14 M. Mulholland and J. Waterhouse, *J. Chromatogr.*, 395 (1987) 539–551.
15 *ESCA (Expert Systems in Chemical Analysis)*, ESPRIT Project number 1570. Project Leader: A. Clelland, Philips Scientific, Cambridge.
16 D. P. Goulder, T. Blaffert, A. Blokland, L. Buydens, A. Chhabra, A. Cleland, N. Dunand, H. Hindriks, G. Kateman, H. van Leeuwen, D. Massart, M. Mulholland, G. Musch, P. Naish, A. Peeters, G. Postma, P. Schoenmakers, M. de Smet, B. Vandeginste and J. Vink, *Chromatographia,* 26 (1988).
17 T. de Marco, *Structured Analysis and System Design,* Prentice-Hall, New York, 1979.
18 H. P. Nii, *Artif. Intell. Mag.*, Summer (1986) 38–53.