3750

TOWARDS
COMPUTER SUPPORTED
MANAGEMENT OF
ANALYTICAL LABORATORIES

J.W.A. Klaessens

**TOWARDS COMPUTER SUPPORTED
MANAGEMENT OF ANALYTICAL
LABORATORIES**

TOWARDS COMPUTER SUPPORTED MANAGEMENT OF ANALYTICAL LABORATORIES

een wetenschappelijke proeve op het gebied van de
natuurwetenschappen

Proefschrift

ter verkrijging van de graad van doctor
aan de Katholieke Universiteit te Nijmegen,
volgens besluit van het College van Decanen
in het openbaar te verdedigen op
dinsdag 18 april 1989
des namiddags te 3.30 uur

door

Joannes Wilhelmus Antonius Klaessens
geboren op 24 februari 1959 te Nederweert

1989

Druk: Reprografie Geneeskunde Faculteit Nijmegen

Promotor: Prof. Drs. G. Kateman

Co-referent: Dr. Ir. B.G.M. Vandeginste

Voorwoord

Het onderzoek dat ten grondslag ligt aan dit proefschrift bestrijkt een periode van vier jaar. Uitgedrukt in manjaren is de omvang echter veel groter. Immers diverse hoofd- en bijvak, oude en nieuwe stijl studenten hebben belangrijke bijdragen geleverd. Hiervoor wil ik dan ook de volgende ("mijn") studenten bedanken: Nico van Buggenum, Joep van Schalkwijk, Theo Saris, Peter Cox, Ludy van Beysterveldt, Paul Mijland, Richard Bezemer en Jan Sanders. De medewerkers van de afdeling Analytische Chemie en alle overige studenten van deze afdeling wil ik danken, niet alleen voor de discussies over het onderzoek, maar ook voor het scheppen van een geschikt werkklimaat.

Voor onderzoek dat betrekking heeft op analytische laboratoria, zijn contacten "in het veld" onmisbaar. Een ieder die hierbij behulpzaam is geweest, ben ik daarvoor zeer erkentelijk. Met name wil ik Bart van de Wijdeven, destijds bij Organon te Oss, danken voor de samenwerking aan het begin van de promotie. Daarnaast ook Edu van Naerssen en Wessel Gademan van het Regionaal Energiebedrijf Dordrecht voor het mogelijk maken om de ontwikkelde software in de praktijk toe te passen.

Tenslotte mag niet onvermeld blijven de dank die ik Ardie van Berkel verschuldigd ben voor de ondersteuning vanuit het thuisfront.

CONTENTS

Part 1: Introduction	1
Towards computer supported management of analytical laboratories	3
Introduction	3
Activities in the laboratory	4
Computer support of data flow control: LIMS	8
Computer support of management	10
Additional LIMS modules for management support	14
References	15
 Part 2: Applied software techniques: digital simulation and expert systems	19
Digital simulation as an aid to sample scheduling in a routine laboratory for liquid chromatography	21
Construction of the simulation model of the laboratory	23
Experimental	28
Application of the simulation model	29
References	33
Problem solving by expert systems in analytical chemistry	37
Introduction	37
Characteristics of expert systems	39
Tools for building expert systems	52
Application of expert systems in chemistry	54
Discussion and future expectations	58
References	61

Part 3: Development of software modules	67
Analysis of laboratory sample input signals	69
Introduction	69
Theory	70
Experimental	76
Results	80
Discussion	91
Conclusion	92
References	93
Program for processing of historical laboratory data based on fuzzy set theory as a tool for decision support	95
Introduction	95
Theory	96
Description of the program FEA	101
Application	104
Conclusions	109
References	109
Expert system for knowledge-based modelling of analytical laboratories as a tool for laboratory management	111
Introduction	111
Systematic modelling of analytical laboratories	113
LABGEN for automated modelling	114
Application of LABGEN: an example	121
Conclusions and future research	127
References	127
Laboratory organization and modelling aspects	129
Application of digital simulation for management support	129
Automatic modelling by LABGEN	131
Simulation model of a simple laboratory organization	133
Simulation and laboratory organization features of LABGEN	138
Discussion	151
References	154

Part 4: Application of software modules to a laboratory for quality control	157
--	-----

Application of FEA and LABGEN	159
Introduction	159
The laboratory situation	160
The simulation project	161
Construction of the model	162
Application of the models	170
Conclusion of the simulation project	177
Evaluation of LABGEN	179
Appendix: model settings of model 1	180
References	185

Part 5: Conclusion	187
---------------------------	-----

Conclusions and general remarks	189
References	191
Summary	193
Samenvatting	195
Curriculum vitae	197

Part 1

INTRODUCTION

TOWARDS COMPUTER SUPPORTED MANAGEMENT OF ANALYTICAL LABORATORIES *

INTRODUCTION

For the control of the value of the information produced by an analytical laboratory, it is not sufficient to consider accuracy and precision of the analytical result [1]. Attention should also be paid to qualifiers such as costs per analysis, sample and data management in the laboratory, delay times of the samples (i.e. the time lag between the sample arrival at the laboratory and the reporting of the result to the customer). As opposed to accuracy and precision which mainly depend on the analytical procedures, the latter qualifiers mainly depend on the organization of the laboratory as a whole. Therefore, the value of the analytical information is also dependent on the organization of the laboratory and on the decision making within it.

Starting from a taxonomy of the activities in a laboratory, this introductory chapter surveys different levels of computer aids to control and improve the performance of the laboratory as a supplier of information. Not only the state of the art is described, but also new software modules for decision support are proposed. The development and application of these modules is the subject of the chapters of the following parts.

* This chapter is part of the paper: J. Klaessens, B. Vandeginste, G. Kateman, Towards computer supported management of analytical laboratories, Anal. Chim. Acta, in press

ACTIVITIES IN THE LABORATORY

A systems representation of an analytical laboratory is shown in Figure 1. The main process, of course, is the production of analytical results. The laboratory operation is surrounded by a control layer, consisting of the laboratory management and administration functions such as reporting to the customer and invoicing. The laboratory should be adapted to the standards and regulations set by controlling agencies, e.g. Good Laboratory Practice (GLP) regulations.

Figure 1: The analytical laboratory as a system.

The model shown in Figure 2 is more appropriate to describe the laboratory management in a more explicit way [2]. Management is con-

Figure 2 Schematic representation of the laboratory management

cerned with decision making, resulting in decisions and instructions for the laboratory, for instance the decision whether or not to purchase a highly expensive instrument, say ICP. In order to arrive at a decision, the manager makes use of his knowledge of and experience with the laboratory (and also analytical chemistry) and his intuition, a valuable good. The decision is ultimately based on the measured performance of the laboratory as compared to a target performance (planned vs. actual) and is influenced by a policy. In the ICP example, the target may be to increase the capacity of the laboratory with respect to heavy metal analyses in order to be able to cope with an expected increase of these samples. The policy may be that in the future no longer only internal samples are to be analyzed, but also samples from external customers. It is this policy which gives rise to the expected increase. The model of Figure 2 does not state to which degree the policy and target emanate from the laboratory management or the higher managerial echelons not represented in the model. This varies very much from laboratory to laborato-

ry as it depends on the size and the structure of the organization the laboratory is part of.

Table 1

Classification of decision problems in the analytical laboratory after Mitra's extension of Anthony's framework [3].

	Objectives	Time horizon	Nature of information	Degree of uncertainty
Strategic planning	resource acquisition	long, ≈ 5 years	highly aggregate	high
Tactical planning	resource utilisation	medium, 0.5 - 1 year	moderately aggregate	moderate
Operations control	optimal lab. execution	month, week, day	detailed	low
Analytical control	optimal chem. execution	day, hour	detailed	low

In order to gain a clear view of possible computer aids for laboratory management, not only a model of the management itself is required but also a taxonomy of decision problems encountered in the laboratory, and the corresponding activities. Table 1 contains a taxonomy based on Mitra's extension of Anthony's framework [3]. Most strategic and many tactical decisions reach at least partly beyond the realm of the laboratory management. However, because the laboratory management has a role in these levels of decision making, it is useful to include these.

An example of strategic planning is already mentioned in the ICP example above: the decision not only to analyse internal samples in the

laboratory but also to offer services to external customers. The purchase of an ICP instrument then is an example of a decision problem of the tactical planning level. Another example is determining the working capacity (number of technicians) required in the laboratory. The activities of the higher levels of Table 1 are non-procedural in nature. These are activities to resolve new problems or to cope with situations not easily described in established procedures.

Moving from the higher to the lower levels of Table 1 the activities gradually become more procedural or routine activities. Crew scheduling is an example of an activity at the level of operations control. This activity is regularly carried out, and is definitely less complex than activities of the higher levels. However, it requires more than the application of a fixed procedure. A considerable amount of skill and intuition is necessary. In a laboratory for quality control, the level of analytical control contains the chemical operations carried out by the laboratory technicians: sample pre-treatment, sample preparation, measurement, calculation of the analytical result. These are mainly routine activities which are regulated by established procedures.

They are accompanied by information exchange activities: gathering information to plan sample scheduling, worklists containing the tests to be performed on each sample, producing reports of the analytical results. Although of a predominantly procedural nature too, these activities can be clearly distinguished from the instrumental and "wet" chemical activities. With respect to the latter group, the technician can be supported by mechanisation (e.g. autosamplers) and, more recently, robotics. Support with respect to the information activities, on the other hand, can almost only be rendered by information technology, i.e. computers.

It is important to point out the difference between information and data. What is information to the technician who is scheduling a sample, is data when it is being archived or retrieved, no longer information. Therefore, only the term data will be used henceforth. Taken all samples together, the result is an extensive flow of data at the analytical chemistry level. Control of this flow is of critical importance for optimal performance of the laboratory.

The activities at the higher levels of Table 1 (referred to as the management level) also require and are based on a flow of information or data. It concerns data about the operation of the laboratory. These are more aggregate than the data at the analytical chemistry level. Information activities at the management level not only consist of data handling, but, more importantly, also of producing the relevant data.

COMPUTER SUPPORT OF DATA FLOW CONTROL: LIMS

Control of the data flows at the analytical chemistry level and the management level are of a different nature and require different approaches for computer support. Control at the analytical chemistry level mainly consists of data handling activities which are straightforward and very frequent. Routine activities are always the first candidates for automation [2]. In many laboratories software packages to take over these data handling activities are already implemented: Laboratory Information Management Systems (LIMS). Several commercial LIMS's are on the market. Although recently a book by McDowall on this subject has been published [4] and the First International LIMS Meeting has been held in Pittsburg [5], relatively few attention has been paid by chemometricians to LIMS and its implications for the laboratory.

In essence, a LIMS is a database for storage of laboratory data, especially data about samples. Depending on the specific implementation, the LIMS has a number of modules that guide the storage and retrieval of data. Modules present in many LIMS's are shown in Figure 3. They are briefly described to show which activities can be taken over by a LIMS and which new activities are introduced.

The first module encountered by an arriving sample is the sample log-in module: all relevant data about this sample are entered into the LIMS, e.g. test assignment to the sample. For routine samples, it is possible to do test assignment automatically. Next in line modules for workload management are encountered. These may perform activities such as: show the progress of sample, produce backlog reports, generate

Figure 3. The LIMS database and some modules which are present in many implementations

worklists for the laboratory staff. With respect to the results of the analytical tests which are performed on the sample, modules can be used for manual data entry, or for automatic data acquisition, e.g. of chromatographic data. Manually entered data can easily be verified whether they have the right format and belong to a range which is chemically allowed. Most LIMS's offer reporting facilities: automatic generation of a report for the customer after all tests are completed and their results are validated. Also modules may be present for inquiries of the database, e.g. to make a trend analysis of the results of a particular sample type.

The major benefits which can be realized by the implementation of a LIMS can be summarised (divided) in four groups. These are:

- 1 Data availability: Once data about a sample entry or analytical results are entered into the LIMS, they are easily accessible for anyone of the laboratory staff who is permitted to. This is advantageous for planning sample scheduling and for result validation.
- 2 Data integrity: The probability of errors is reduced because less data transcription is needed, automatic data acquisition can be employed, manually entered data can be controlled.

- 3 Data manipulation: Calculation of the analytical result can be performed by the LIMS.
- 4 Productivity: Because the LIMS takes over several data handling activities such as calculating the result and reporting, an increase of productivity will be the result.

In order to realize these benefits, the implementation must be carefully planned and carried out. A successful implementation requires an integration of the computer in the laboratory organization. Therefore, it would be very interesting to investigate the possibility of using the LIMS as an environment for chemometric software tools such as experimental design, curve resolution, expert systems for chemical advice.

COMPUTER SUPPORT OF MANAGEMENT

Computer support at the management level is different from support at the analytical chemistry level, because producing the relevant data (information) is important, rather than handling the data. This is rather complicated. For instance, which data are necessary for an optimal decision about the purchase of an ICP instrument? It is generally expected that at this level the computer will aid the human being rather than take over some of his activities [2]. Although some relevant modules are already available in LIMS (e.g. for database inquiries), at present the computer is only beginning to be applied for this purpose. This is closely related to the development of expert system programming techniques. Several reviews on expert systems have already been published in the chemical literature [e.g. 6,7].

The data stored in a LIMS in fact reflect the character of the laboratory in the way it responds to changes in the sample input stream and the demands of its environment. As a consequence, application of these data to solve management problems may be very useful. However, it must be realized that the data stored in a LIMS are not suitable to be immediately used for this purpose. It is necessary to have software modules to refine the data into a suitable form.

With respect to these software modules, two major areas of management support can be distinguished: support of short term management at the level of operations control (Table 1) and support of long term management at the levels of tactical and strategic planning. The first consists of computer aid in sample routing and crew scheduling. The current expert system techniques probably allow the development of a computer system which takes over these activities to a great extent. Such an expert system should combine the information about the workload (i.e. the active samples stored in the LIMS) with knowledge about the way the laboratory reacts to changes in input and resources. This knowledge is very specific, and is continuously changing. Therefore, it must be doubted whether such an expert system is cost-effective in small and medium sized laboratories. A computer aid which supports the manager by making possible a quick and easy access to the relevant data can be realized by the current LIMS techniques.

Actual take over by the computer of long term management activities cannot be realized yet, probably because the uncertainties involved are too high. The support will merely consist of extending the capabilities of the human manager. Figures 4 and 5 show two mechanisms to accomplish this. The first mechanism (Figure 4) implies the development of software modules producing a representation of the data which is more informative towards decision making. Many LIMS's already contain modules for this purpose: modules for data inquiries, e.g. to make a trend analysis of the test results of a particular sample type or of their delay times. However, the tasks offered reflect rather the possibilities of the current database techniques than what is relevant from a management point of view. A major drawback is the employment of strict relations: retrieve all sample entries with some specific property. For management purposes it is often more important to be able to study groups of sample entries which are related with respect to a property, rather than being identical. For instance, the question "How well does the laboratory perform with respect to heavy metal analyses?" will usually be more important than "How well does it perform with respect to Cd analyses?". A second major drawback is that it is difficult and time consuming to gain a general view of the performance of the laboratory. Most

Figure 4 Possible mechanism for computer support of long term management of analytical laboratories without simulation

managers will not find enough time for a thorough analysis. Supplementary representation methods are necessary.

An alternative method of management support can be provided by the development of modules which make possible the evaluation of proposed decisions before they are actually carried out. Likely, modules for decision evaluation based on digital simulation will prove very effective [8,9,10]. As is shown in Figure 5, this method also requires a first step of processing the laboratory data of the LIMS. However, the steps between data processing and the decision are made more explicit. Data are incorporated into a simulation model and alternative decisions can be objectively evaluated by means of simulation experiments. A model of the laboratory management employing digital simulation is shown in

Figure 5 Possible mechanism for computer support of long term management of analytical laboratories with simulation

Figure 6. As compared to Figure 2, a simulation loop is added between two switches. Proposed decisions and instructions are incorporated into the simulation model and the simulation results are compared with the planned results (target) If satisfactory, the decisions and instructions are applied to the real laboratory.

Figure 6 Model of the laboratory management employing digital simulation as a decision support tool

ADDITIONAL LIMS MODULES FOR MANAGEMENT SUPPORT

Because many relevant data are stored in the LIMS, it is suitable to incorporate the modules for management support into this environment. An important constraint is that the simulation modelling and experimentation can be conducted by the laboratory staff within a reasonable short time and without external expertise required. In order to meet this constraint, software modules are necessary that fully guide the simulation project. They should require input and render output which can easily be understood by the manager.

Bearing this in mind, software modules have been developed for:

- processing of historical laboratory data;

- automatic construction of simulation models of analytical laboratories.

The structure and the application of these modules are discussed in the following chapters.

In the following part consisting of two chapters the most important software techniques which are used for the development of the modules are described - digital simulation and expert systems. Considerable interest has been paid by the simulation community to the application of artificial intelligence techniques to simulation [11,12]. Research papers have been published on the development of intelligent environments for model construction for continuous flow simulation [13,14] and discrete event simulation [15-20].

In two following parts respectively the actual development and the application of the software modules are discussed. The modules are applied in a laboratory for quality control of drinking water.

REFERENCES

- 1 B.G.M. Vandeginste, *Analisis*, 12 (1984) 496
- 2 J.A.G. Hale, Goal oriented decision support, in: G. Mitra (ed.), *Computer assisted decision making*, Elsevier, Amsterdam, 1986, p. 35
- 3 G. Mitra, Introduction and overview, *ibidem*, p. ix
- 4 R.D. McDowall, *Laboratory Information Management Systems*, Sigma Press, Wilmslow, 1987
- 5 R. Megargle, R. Wilkes, *Trends Anal. Chem.*, 6(10) (1987) xi
- 6 R.E. Dessy (ed.), *Anal. Chem.*, 56 (1984) 1200A, 1312A
- 7 J. Klaessens, G. Kateman, *Fres. Z. Anal. Chem.*, 326 (1987) 203
- 8 T.A.H.M. Janse, G. Kateman, *Anal. Chim. Acta*, 159 (1984) 181
- 9 P. Winkel, *Clin. Chem.*, 30 (1984) 1758
- 10 B. van de Wijdeven, J. Lakeman, J. Klaessens, B. Vandeginste, G. Kateman, *Anal. Chim. Acta*, 184 (1986) 151
- 11 M.S. Elzas, Relations between artificial intelligence environments and modelling & simulation support systems, in: M.S. Elzas, T.I. Oeren, B.P. Zeigler (eds.), *Modelling and simulation methodology in the artificial intelligence era*, Elsevier, Amsterdam, 1986, p. 61
- 12 R. O'Keefe, *Simulation*, 46 (1986) 10
- 13 R. Muetzelfeldt, A. Bundy, M. Uschold, D. Robertson, ECO - An intelligent front end for ecological modelling, in: E.J.H. Kerckhoffs, G.C. Vansteenkiste,

- B P Zeigler (eds), AI applied to simulation, Simulation series vol 18 nr 1, SCS, San Diego 1986, p 67
- 14 P A Luker, Putting expertise into Modeller, *ibidem*, p 103
 - 15 B Khoshnevis, An-PIN Chen, An expert simulation model builder, in P A Luker, H H Adelsberger (eds), Intelligent simulation environments, Simulation Series vol 17 nr 1, SCS, San Diego, 1986, p 129
 - 16 K E Wichmann, An intelligent simulation environment for the design and operation of FMS, in J E Lenz (ed), Proceedings of the 2nd International Conference on simulation in manufacturing, Chicago 1986, p 1
 - 17 N R Nielsen, Knowledge-based simulation programming, in A Mattox (ed), AFIPS Conference Proceedings Vol 55 1986, AFIPS Press, Reston. 1986, p 125
 - 18 G I Doudikis, R J Paul, Experiences in automating the formulation of discrete event simulation models, *ibidem* [15], p 79
 - 19 F J Wales, P A Luker, Design of a multi-level user interface for discrete event simulation, in G C Vansteenkiste, E J H Kerckhoffs, L Dekker, J C Zuiderv-aart (eds), Proceedings of the 2nd European Simulation Congress, SCS, San Diego, 1986, p 361
 - 20 A Lehman, B Knoedler, E Kwee, H Szczerbicka, Interactive modelling and simulation in an intelligent PC-environment, *ibidem*, p 347

Part 2

APPLIED SOFTWARE TECHNIQUES: DIGITAL SIMULATION AND EXPERT SYSTEMS

DIGITAL SIMULATION AS AN AID TO SAMPLE SCHEDULING IN A ROUTINE LABORATORY FOR LIQUID CHROMATOGRAPHY

BART VAN DE WIJDEVEN ^{AND} JAAP LAKEMAN

Organon International B V, Kloosterstraat 6, 5349 AB Oss (The Netherlands)

JO KLAESSENS*, BERNARD VANDEGINSTE and GERRIT KATEMAN

*Laboratorium voor Analytische Chemie Katholieke Universiteit Nijmegen Toernooiveld
6525 ED Nijmegen (The Netherlands)*

(Received 27th December 1985)

SUMMARY

A computer model is described to aid decision making in a routine laboratory for liquid chromatography. The managerial demand was to reduce the delay times of the samples in the laboratory despite an increasing sample input. Knowledge derived from records on the laboratory and from the experience of the laboratory staff is used to develop a simulation model of the laboratory. The model is applied to study the effects of several factors on the performance of the laboratory in order to trace the bottlenecks.

The value of analytical information depends on four factors: precision, accuracy, result delay and sampling planning [1]. The chemical or instrumental procedure remains vital, as precision and accuracy largely depend on it. However, because chemical analysis is essentially part of a decision-making process with regard to products or processes, it becomes important to evaluate what information is needed for a given purpose and at what cost. The analytical chemist is an expert who translates the client's problem into analytical constraints (the four factors mentioned above), gathers the optimal amount of information, converts it into common language and finally helps the client to solve the problem.

The result delay is the time lag between the sample entering the laboratory and the result becoming available and is usually governed by the organization of the laboratory. In studies of the operation of routine laboratories, it was found that it is not exceptional that samples requiring only a few hours of working time can remain in the laboratory for weeks [2–4]. Too large a delay time will decrease the value of the result, or even render it worthless.

For instance, if it is considered that, in order to control a chemical process, it has to be monitored more accurately, one would be inclined to increase the sampling frequency. However, the input to the laboratory will then increase and so will the delay times for results. If the laboratory already has a high workload, the increase in delay will be dramatic. The later the analytical

result becomes available, the less valuable it will be for process control. In order to improve the monitoring of the process, the bottlenecks in the laboratory organization must be traced and removed.

Description of the problem

A routine laboratory can be a rather complex structure governed by implicit and explicit rules. The laboratory to be considered here is an independent section of a quality control laboratory for a producer of pharmaceuticals; liquid chromatography is done by four technical staff. The input of the section is rather complicated because there are four sources of sample, each of which has its own priorities: (1) from production, samples are taken from a batch of products which can only be released after a pharmacist has approved on the basis of the analytical results; (2) samples from a research department; (3) samples from a microbiology department; (4) samples for testing the shelf-life of pharmaceutical products. Most samples originate from production (Table 1). Every morning, the head of the laboratory decides on the samples that are to be analyzed that day, i.e., the workload. A technician takes a sample from the workload and runs the analysis. All four technicians are able to process all types of samples, and, as a consequence, to operate all h.p.l.c. instruments. In the laboratory, a priority rule is applied which is based on the delay times of the samples. In addition, production samples have priority because of their direct commercial consequences; batches of products must be stored until the analytical results become available.

Although the laboratory was facing an increasing sample input, a reduction in delay time (especially of production samples) was desired. Several solutions were possible: more equipment, increase of manpower, change in priority rules, and prior warning of the expected arrival times of the samples which would improve the planning within the laboratory. Because of lack of knowledge about the factors that govern the operation of the laboratory, these alternatives could not be evaluated. However, some knowledge was available: historical data on the laboratory and the experience of the laboratory

TABLE 1

Annual sample input (in sample units, total input of the real laboratory is 100), real laboratory (June 1982–May 1983) and five simulation runs with different random numbers

	Stability	Microbiol	Research	Production	Total lab
Real	7.8	4.2	26.1	61.9	100.0
Run 1	6.3	3.9	24.7	61.5	96.4
Run 2	1.6	4.3	30.7	62.1	101.8
Run 3	7.1	3.2	29.7	67.7	107.7
Run 4	4.9	4.3	30.1	65.9	105.2
Run 5	6.1	4.4	29.0	68.8	108.5

staff and personnel. From these sources, a decision support system based on digital simulation was designed. The former type of knowledge must be viewed in relation to commercially available laboratory information management systems (LIMS) [5] which will make the historical data more easily accessible.

CONSTRUCTION OF THE SIMULATION MODEL OF THE LABORATORY

A vast literature on simulation and many textbooks [e.g., 6, 7] are now available. However, only a few applications to clinical laboratories [8, 9] and analytical laboratories [2- 4, 10] have been reported. These reports mainly investigate the suitability of digital simulation to describe an analytical laboratory [2] and the impact of various factors on the performance of the laboratory [3, 4, 10]. In the research reported here, the managerial objectives are easily discerned. Because of lack of well structured knowledge, it was decided to design a simulation model of the laboratory.

In the application of digital simulation to problem solving, three phases can be discerned: (1) the modeling phase in which a computer model of the laboratory is constructed; (2) model verification and validation; (3) the experimental phase in which simulation experiments are conducted to support decision making.

The modeling phase: description of the simulation model

An analytical laboratory can be defined as a network of service stations with an input of samples. During the modeling phase, both the network (structure) and the input are translated into a simulation model that approximates reality in its stochastic properties.

In general, a model consists of classes of objects and queues. A class contains rules and attributes and a specific object of this class is created by assigning values to these attributes. For instance, an object of the class "sample" can be specified by its origin, arrival time, due time, working time, etc. Objects can be stored in queues from which they are retrieved when some operation is about to be done, e.g., processing of a sample. To some extent, both translation processes can be considered separately.

In this study, the input generating part of the model was defined as a class of generators, consisting of four objects: the sample sources as previously mentioned. In order to specify the four generators, the real input of the laboratory was analyzed from filed historical data. From these files, approximately 100 different types of sample, characterized by the chromatographic procedure, were discernible. This number is too large to handle efficiently in the model and the low frequencies of many types make it rather unstable. Therefore a cluster analysis [2] was done in close cooperation with the laboratory personnel. This produced a reduction to 18 types, e.g., by combination into one new type of several types that required similar sample preparation and apparatus.

In order to quantify the generators, the main statistical characteristics of the real input were determined, i.e., distributions of input samples according to origin and type, and their mean values and standard deviations. In the model, each generator has a typical distribution of entries per day from which a random value is taken for each morning. Then, the type of every entry is established, also randomly according to a distribution that is specific with respect to the generator. It appeared that some types may enter the system in batches. So, for every entry the batch-size is evaluated according to a distribution that is specific with respect to generator and type. An entry may consist of one sample or of a batch of samples. Table 2 contains the relative input frequencies of the sample types of the production generator. The structure of the generating part of the model is depicted in Fig. 1.

The next step is the translation of the structure of the laboratory. The main building blocks are: a class of samples, a class of liquid chromatographs, a class of technicians and a planner. The planner contains the rules that control the sample throughput; to some extent it is the base of rules for the system. Many of its parameters cannot be reconstructed from the record files and must be estimated in close cooperation with the laboratory personnel. This also applies to many parameters of the other objects in the model. For instance, the preparation times of all sample types and their standard deviations were estimated in a session with the laboratory personnel. From these estimates, uniform distributions of preparation times were created, from which a random value was taken for every sample. Other parameters are well defined, e.g., chromatographic times and injection times.

Several queues are used in the model to store the samples before the different processes (Fig. 2). After generation, the samples are stored in a queue called shelf. Every morning, the planner takes the newly entered samples from the shelf and tries to combine samples of the same type into batches for which the total batch preparation time does not exceed 7 h. Treatment in batches reduces the total processing time in routine work. These combinations concern samples that enter the system simultaneously; they can always be combined if they have the same type; only the total preparation time serves as a restriction. Next, the planner tries to combine

TABLE 2

Production generator (model) types of generated samples and their relative frequencies

Type	Relative frequency	Type	Relative frequency	Type	Relative frequency	Type	Relative frequency
1	0 10	6	0 09	11	0 04	15	0 03
2	0 08	7	0 07	12	0 05	16	0 09
3	0 09	8	0 07	13	0 03	17	0 03
4	0 11	9	0 03	14	0 06	18	0 01
5		10	0 03

Fig. 1 Model of the input generators.

Fig. 2 Sample stream through the laboratory and the hierarchy of the queues (in rectangular boxes)

Fig. 3 Flow sheet of the activities of a technician; work out (1) asks if the total work-out time is longer than 5 h

the newly entered samples with samples in the work queue that entered the system earlier. Combination of samples of the same type is not always possible at this stage. This was approximated in the model by the introduction of a so-called combination chance. The samples (or batches of combined samples which will also be referred to as samples) are placed in the work queue and ordered according to due time. The planner looks at all samples in the work queue to see if they should be in the urgency queue. Samples to be moved to the urgency queue are mainly production samples whose due time will elapse within a certain period ("urgency factor" days). Apart from these, there are urgent samples which enter the laboratory having short due times, and which may originate from all four generators. They must always be placed in the urgency queue. This reflects the real situation in which information is quickly needed, e.g., to anticipate a halt in a production line.

The planner's next job is to fill the scheduling queue which contains the samples that are to be processed, i.e., the workload of that day. Samples are assigned to all chromatographs that are available or will become available during the day. As many urgent samples as possible must be assigned and the samples will not be suitable for all instruments (varying from 1 to 5), thus establishing the workload is a rather complicated procedure. The technicians take the samples from the scheduling queue as required.

Compared to the planner where many implicit rules had to be stated explicitly, the other objects in the laboratory are relatively simple. A class of six h.p.l.c. instruments was defined. The objects of this class have rules only with respect to three states (active, passive and busy). The class covering technicians contains more activities. There are four technicians, one of whom is working in the model only when the workload is high. In reality, this fourth technician is called from elsewhere at need. The model has an advanced time handling mechanism and the simulation covers all day: the technicians start at 8.00 a.m. and work for eight hours a day. For simplicity, weekends are discarded. The job of the technicians consists of sample preparation and tuning of the chromatographs which actually run overnight. Then, the results must be calculated out and checked, a report is written. If the workload is high, these jobs accumulate until a work-out time limit has been exceeded. Figure 3 contains a flowsheet of the activities.

The model is completed with two disturbance processes in order to simulate the breakdown of an instrument. In case of the first kind of disturbance, immediate correction by the technician is possible after which the running job will be started from the beginning. It models the technician's decision from a first examination that the chromatography must be repeated because of some failure although the instrument did not break down. In the second case (breakdown of an instrument), immediate correction is not possible, the running job is rescheduled and the instrument becomes available again after a randomly distributed disturbance time.

Only a rough description of the simulation model has been given. Many details of the model are not mentioned because they are not essential for understanding the model.

Model verification and validation

A simulation model must gain credibility before it can be used to support decisions, i.e., it must be subjected to a test procedure which involves several steps [11]. Before the construction of the computer model is started, the plausibility of the model must be checked by the laboratory staff who know the organization. Other verification steps involve checking of historical data and checking whether the program runs as planned.

The next stage concerns validation of the model, i.e., its ability to predict the laboratory output if it is fed with input that shows the same statistical characteristics as the real input. Therefore the input generating part of the model must be tested first. Although there is some disagreement about the method to be used, the method of hypothesis testing is commonly adopted, i.e., the null hypothesis H_0 that there is no significant difference between model and reality. Five different simulation runs with different random numbers were used. The total sample input from the four generators is listed in Table 1. Figure 4 shows the real and one of the simulated input distributions of the total laboratory. The χ^2 test ($\alpha = 0.05$) indicated that none of the simulated distributions differed significantly from the real distribution. Autocorrelation functions of the real and simulated input showed that they are not significantly correlated in time.

After the input has been tested, the laboratory performance can be tested by comparing the real and simulated output. Usually, a model contains several parameters that are difficult to estimate beforehand. For instance, in the case of a high workload, a fourth technician is added to the laboratory; it is very difficult to express precisely the criterion to be used. Initially, there will be no agreement between simulated and real output because of these vague parameters, which must be tuned gradually until both outputs match. This process is called model calibration. Only the end results of the calibration are shown below. The simulation was run five times with different random numbers, so as to obtain statistical information about the behaviour of the model. Figure 5 shows the distributions of the delay times of the samples in

Fig. 4. Input distributions (% frequency vs sample units) for the total laboratory: (a) real; (b) simulated.

Fig 5 Distributions (% frequency vs. time units) of the delay times for the total laboratory. (a) real, (b) simulated.

the real and the simulated laboratory; they resemble each other fairly well. The mean values of the delay times and the standard deviations are shown in Table 3; the delay times are correlated in time which may reduce the standard deviation. Therefore, the standard deviation was also calculated from the five simulation runs. Normal distribution of mean delay times was assumed. Because the real mean delay time belongs to the interval $\bar{x}_m \pm ts$ ($\bar{x}_m$ and s are the mean value and standard deviation of the simulation runs, respectively, and t is Student's t for $\alpha = 0.05$ and d.f. = 4), there is no significant difference between simulated and real delay times, except for microbiology. The real value for microbiology is affected by some processing errors in the laboratory. As shown in Table 1, microbiology constitutes only a small part of the input and the difference will hardly affect the total performance. Some other factors were used to test the model, i.e., percentage of samples overdue, availability of instruments, mean number of combined samples, and the daily working times of the staff (Table 4).

From these results, reality and model coincide fairly well. Further refinement would be possible but the loss in clarity of the model and the extra cost would probably not compensate for the gain in reliability. The extent of refinement is not governed by what is possible but by what is optimal for managerial purposes. Even before any experiments were done, the simulation approach provided useful information on which parameters were important with regard to laboratory performance. However, better insight was gained from experiments.

EXPERIMENTAL

The model of the laboratory was written in PROSIM [12] which is a special simulation language based on PL/I. The program was run on an IBM 3081 mainframe computer. Simulation of the annual operation of the laboratory took about 30 cpu seconds.

TABLE 3

Mean delay times during the period 1982–1983 (in time units; the mean real delay time is 100) for real laboratory and five simulations. Standard deviations are given in parentheses

	Stability	Microbiol.	Research	Production	Total lab.
Real	258 (183)	20 (43)	147 (145)	66 (45)	100 (114)
Run 1	144 (150)	15 (6)	148 (116)	62 (31)	86 (82)
Run 2	278 (270)	16 (9)	184 (177)	68 (35)	107 (129)
Run 3	297 (232)	14 (6)	188 (142)	62 (33)	108 (121)
Run 4	282 (239)	16 (8)	160 (153)	62 (33)	94 (110)
Run 5	97 (115)	15 (8)	190 (219)	68 (40)	97 (128)
Mean runs	219 (92)	15.3 (1.0)	174 (19)	67 (3)	103 (9)

TABLE 4

Comparison of the performance of real laboratory and five simulations during 1982–1983

	Overdue ^a (%)	Machine av. ^b	Comb. ^c	Daily working hours ^d			
				1	2	3	4
Real	28.5	60–70	2.4	—	—	—	—
Run 1	21.4	61	2.1	6.39	6.58	6.34	2.78
Run 2	13.7	68	2.2	6.45	6.28	6.30	4.58
Run 3	25.8	66	2.4	6.37	6.45	6.35	4.31
Run 4	22.5	63	2.2	6.53	6.52	6.31	3.23
Run 5	24.1	66	2.3	6.24	6.20	6.31	3.26

^aPercentage of samples with delay times that exceed the planned due time. ^bMean availability of the instruments (%). ^cMean number of samples combined into batches. ^dFor technicians 1–4.

Historical data were used from the period June 1982–May 1983. A file of magnetic tape contained the arrival times and report times of all samples. A file of sample documents contained the characteristics for the samples: (a) number of independent determinations; (2) preparation time (e.g. an extraction, if needed, takes about 1.5 h); (3) type of chromatograph with its particular mobile phase; (4) number of injections per sample; (5) run time per injection. In addition to these data, many important characteristics of the laboratory were estimated in sessions with the laboratory staff.

APPLICATION OF THE SIMULATION MODEL

Conducting experiments

Experiments with a simulation model must be conducted very carefully. Essentially, the model can be no more than an approximation of the real

laboratory with regard to sample throughput. Because the validation was restricted to a limited time period and no generality in input was pursued, the model has only a limited reliability. Consequently, exploratory experiments can be trusted only if the effect of slight changes in the parameters is measured. These changes should not lead to a situation that is too far removed from the validated structure. However, careful experimentation within these limits will reveal the importance of the variables of interest.

The influence of some variables often depends on the level of the other variables. For instance, the effect of adding a new instrument will be negligible if the number of technicians is the real limiting factor, whereas it can be important if the system retains some flexibility with respect to manpower. With a minimal number of experiments, the interactions between variables can best be measured by means of experimental (or factorial) design. Extensive discussions of experimental designs, along with methods for constructing and evaluating the designs, were given by Davies [13] and Kleijnen [14].

Results of the experiments

In addition to the number of technicians and chromatographs, some other factors that give a better insight into the performance of the system were evaluated: (1) the due time of production samples (i.e., the major part of the input), (2) the fraction of urgent samples, i.e., generated with smaller due times (uniformly distributed from 2 to 10 days), (3) the workload factor (the workload for the next seven days is calculated every day, and if it exceeds the workload factor the fourth technician is called in and stability samples are no longer analysed), (4) the time needed to work out the results of a chromatographic run and to write the report, (5) repetition factor, i.e., the fraction (%) of repetitions of chromatographic jobs, which controls the first of the above-mentioned disturbance processes, (6) the mean inter-disturbance time, i.e., the mean value of the negative exponentially distributed time between two instrument breakdowns, and (7) the urgency factor for which the samples are moved from the work queue to the urgency queue if the due time will elapse within a period of "urgency factor" days.

The main effects of these seven factors and some of their interactions can be measured by using a factorial design of 16 experiments. In order to obtain statistical information, each experiment was repeated. The high and low levels of all factors are listed in Table 5. The performance of the system was measured by means of the following entities: the delay times of research and production samples and of the total laboratory, and the total busy time of all technicians per day. The significant effects are listed in Table 6, none of the interactions appeared to be significant. The most important effect is clearly the work-out time. This is as expected, longer work-out times per sample will result in longer busy times. There is also an effect on the delay times which is larger for research samples because production samples have priority. The data indicate that the system is sensitive to changes that concern the workload of the technicians. A change in the urgency factor does

TABLE 5

Levels of the factors in the factorial design as compared to the normal values

Level	High	Low	Normal
Due time production (t u)	80	40	60
Urgent sample fraction (%)	20	3	10
Workload factor (h)	180	80	130
Work out time (units)	1 5	0 5	1
Repetition factor (%)	25 5	4 5	16
Mean int. dist. time (t u)	20	10	15
Urgency factor (t u)	15	5	10

TABLE 6

Results of the factorial design

Level	Delay times (t u)			Busy times ^a (h)
	Research	Production	Lab	
Due time rel.	—	19 ^b	—	—
Urgent sample fraction	-66 ^c	—	-18 ^d	—
Workload factor	48 ^d	—	18 ^d	-0 7 ^c
Work out time	98 ^b	15 ^c	31 ^b	3 8 ^b
Repetition factor	—	—	—	—
Mean int. dist. time	—	9 ^e	—	0 5 ^d
Urgency factor	—	—	—	—
St. dev. effects	22	5	14	0 46

^aMean working time per analyst. Student's *t* test (d.f. = 16), ^bsignificant $\alpha < 0.005$, ^csignificant $0.005 < \alpha < 0.01$, ^dsignificant $0.01 < \alpha < 0.05$, ^esignificant $0.05 < \alpha < 0.1$.

not cause a change in the workload, it appears to have no effect. Increased frequency of repetition obviously increases the workload somewhat, although no effect was measured. A higher frequency of machine breakdowns increases the workload more because the sample preparation has to be repeated. Indeed, some effect was measurable. The effects of the urgent sample fraction can be understood, because research samples normally have low priority but an increase of urgent samples with high priority reduces the mean delay time. Finally, the workload factor controls the activity of the fourth technician and therefore an increase of this factor causes a decrease of the total busy time of the technicians and an increase in the delay times. This factor, however, will strongly affect the delay times of stability samples, the effect was not measured.

All effects support the conclusion that the system is sensitive to changes in workload or, correspondingly, working capacity, and that other factors are less important. To confirm this conclusion, a new factorial design was

tested, a 2^2 design with two factors (number of technicians and number of instruments). The low levels of both factors are equal to the laboratory situation during 1982–1983. The high levels are an extra allround technician and an extra chromatograph. The measurements were made in duplicate. The significant effects are listed in Table 7. This design strongly indicates that a lack of manpower is the real bottleneck in the system.

The same experiments were conducted using the validated model for the period June 1983–May 1984. For this validation, only the input generation had to be adapted to a larger workload. The old and new inputs are compared in Table 8. In spite of the larger input, almost the same factors showed significant effects. For instance, in the 2^2 design with the factors technicians and instruments, only the effect of the number of technicians was significant, this time even more so because of the increased workload.

Conclusion

The most important conclusion to be drawn from the simulation experiments is that the managerial aim to reduce the delay times can best be met by increasing the working capacity of the laboratory. This conclusion was handed over to the laboratory management. The impact of adding extra personnel to a small laboratory is very large, and one must be careful not to

TABLE 7

Significant effects of the 2^2 factorial design (1982–1983); Student's t test ($df = 4$)

	Delay times (t u)		
	Research	Production	Lab
Technicians (a)	–92 ^a	–22 ^b	–42 ^b
Instruments (b)	—	—	—
Interaction (a–b)	—	—	—
St dev effects	19	6	11

^aSignificant $0.005 < \alpha < 0.01$, ^bsignificant $0.01 < \alpha < 0.05$

TABLE 8

Comparison of the performance of the real laboratory during 1982/1983 and 1983/1984

Factor	Difference (%)	Factor	Difference (%)
Total input (samples)	6.9	Overdue ^a	15.4
Delay times research	15.5	Samples in lab	17.0
production	3.8	Technician hours per day ^b	6.9
lab	7.4		

^aThe percentage of samples with delay times that exceed the planned due time. ^bResults obtained from simulations

depart too far from the validated model. This would decrease the reliability of the conclusions. Moreover, a very extensive reorganization would not gain the support of the laboratory personnel, which is essential for success.

The best plan is to weigh the cost of extra staff against the profit resulting from the calculated reduction of due times of production samples, and to determine the optimum. In this case, it was optimal to add an extra technician, which could be on a temporary basis. After a while, the new situation can be evaluated to check whether the forecast improvement is real and a permanent settlement may then be established.

The construction of a simulation model is a time- and money-consuming activity. It took about six months to model the laboratory and conduct the experiments. In some applications, the investment would not be profitable if the simulation model were used only once. Mostly, however, the model is easily kept up to date and can be used for decision support over a long time.

Two recent developments may be important with respect to computer-aided decision-making based on digital simulation. First, application of artificial intelligence techniques [15, 16] can facilitate the construction of the model. The second important development is the introduction of a LIMS, which would make historical data more easily accessible and would facilitate the input and analysis. A LIMS offers only poor facilities with regard to decision-making as it cannot evaluate managerial proposals. Implementation of modules for expert modeling and simulation into a LIMS context will offer these facilities.

The authors are grateful to Mr. Rob Sierenberg (Sierenberg & de Gans B.V.) for his professional expertise in the use of PROSIM and his suggestions during the design and construction of the model. We are indebted to Mr. Chris Jansen (Organon International B.V.) for his valuable assistance and helpful comments throughout the project, and also to Mr. Henk van Zwam (Organon International B.V.) for the computer calculations and data reductions.

REFERENCES

- 1 B. G. M. Vandeginste, *Analyst* 12 (1984) 496.
- 2 J. A. H. M. Janse and G. Kateman, *Anal. Chim. Acta* 159 (1984) 181.
- 3 B. G. M. Vandeginste, *Anal. Chim. Acta*, 112 (1979) 253.
- 4 B. G. M. Vandeginste, *Anal. Chim. Acta*, 122 (1980) 435.
- 5 R. F. Dessy, *Anal. Chem.* 55 (1983) 70A, 277A.
- 6 J. A. Payne, *Introduction to Simulation: Programming Techniques and Methods of Analysis*, McGraw Hill, New York, 1982.
- 7 A. M. Law and W. D. Kelton, *Simulation Modeling and Analysis*, McGraw Hill, New York, 1982.
- 8 I. K. Vaananen, S. Kivirikko, J. Koskenniemi and A. Relander, *Methods Inf. Med.*, 13 (1974) 158.
- 9 B. Sandblad, in M. H. Hamza (Ed.), *Proceedings of the International Symposium and Course Simulation 1975*, Acta Press, Calgary, Zurich, 1975, p. 364.
- 10 J. G. Vollenbroek and B. G. M. Vandeginste, *Anal. Chim. Acta*, 133 (1981) 85.

- 11 B Page, *Angew Inf Appl Inf* , 25 (1983) 149
- 12 R W Sierenberg, in L Dekker, G Savastano and G C Vansteenkiste (Eds), *Simulations of Systems '79*, 9th IMACS Congress Sorrento 1979, North Holland, 1980, p 997
- 13 O L Davies (Ed), *The Design and Analysis of Industrial Experiments*, Oliver and Boyd, Edinburgh, 1956
- 14 J P C Kleynen, *Statistical Techniques in Simulation*, Dekker, New York, 1975
- 15 J W A Klaessens, G Kateman and B G M Vandeginste, *Trends Anal Chem* , 4 (1985) 113
- 16 S A Borman, *Anal Chem* , 57 (1985) 983A

PROBLEM SOLVING BY EXPERT SYSTEMS IN ANALYTICAL CHEMISTRY*

ABSTRACT

It is expected that artificial intelligence and more particularly, expert systems, will have a profound effect upon analytical chemistry. This paper provides an overview of this rapidly evolving field. General aspects of expert systems are dealt with first, such as knowledge representation, knowledge manipulation, dealing with uncertainty, and the application of software tools to facilitate the construction of expert systems. The accomplishments with respect to the application of expert systems to chemistry - particularly analytical chemistry - are discussed. Future directions of the research and applications are forecasted.

INTRODUCTION

An expert in AAS will encounter considerably less difficulty in analysing a non-routine sample by AAS than a chemist having no experience with AAS. The former has knowledge at his disposal: he knows which information is necessary with respect to the sample preparation and measurement conditions. He can obtain this information from the literature in a reasonably short time, and will easily understand this information. While conducting the experiments, the expert will be able to use vague knowledge which is theoretically not well founded, e.g. rules of thumb. These rules do not have general applicability but the expert can tell from his experience whether or not to apply one in a giv-

* This chapter is part of the review paper: J. Klaessens, G. Kateman, Problem Solving by Expert Systems in Analytical Chemistry, Fres. Z. Anal. Chem., 326 (1987) 203-213

en situation. Indeed, it is his experience which enables the expert to handle problems more efficiently, and which makes his expertise valuable. Expertise cannot be derived from textbook knowledge only, it must be acquired during a long and troublesome learning process.

Frequently, a non-expert wants some expert advise in accomplishing some task, or an expert wants to evaluate his solution for some problem. For example, a chemist with only little experience in AAS has to carry out an analysis with this technique. Without expert advise, considerable preliminary work has to be done before the actual analysis can be started. Because an expert is not always available and always expensive, some other means for expert advise would be very useful.

Recent developments in Artificial Intelligence (AI) indicate that the computer will be likely to become very important in this area. AI is concerned with designing computer systems that to some extent emulate some of the characteristics of human thought: the ability to learn, reason and solve problems. Initially - more than twenty years ago - AI research particularly aimed at designing general and powerful techniques for solving many types of problems. A product of this research is Newell's General Problem Solver GPS [1]. It successfully solved small problems such as puzzles. However, it could not tackle more realistic problems. Its problem solving method was too general to efficiently search a vast space of possible solutions.

More recently, however, AI programs using less general problem solving methods appeared to be more successful. These programs, which are generally called expert systems, are designed for specific domains, and can, as a result, be adapted to the properties of the domain. Two of the most well-known expert systems are MYCIN [2] for medical diagnosis and DENDRAL [3] for structure elucidation of organic compounds. Although both systems are rather successful, using the problem solving method of one system to solve a problem in the other's domain would not yield any positive result. The problem solving methods are too specific.

An expert system not only contains a problem solving method but also knowledge of its domain. An expert system for advise in AAS, for instance, would contain knowledge to decide which lamp must be used

for the determination of a particular element and so on. The system's success critically depends on this knowledge. Only if it is of expert level, the system may be able to give expert advice with the advantage, however, of not overlooking important factors the human expert is inclined to. Because of their ability to manipulate knowledge, expert systems can be applied in many different scientific fields. This is reflected in the explosively increasing interest in expert systems in very diverse fields such as chemistry, medicine, simulation [4] and administration [5].

In this paper we will discuss more general aspects of expert systems firstly, such as knowledge representation, knowledge manipulation, dealing with uncertainty, and the application of tools to facilitate the construction of a new system. Secondly, the expert systems with a chemical domain will be reviewed. Finally, we will try to explore what might be the impact of expert systems on analytical chemistry.

CHARACTERISTICS OF EXPERT SYSTEMS

There are differences between expert systems and conventional programs. The latter have a fixed structure, and the instructions are carried out by the computer in a fixed and predictable order. The line of reasoning must be written out beforehand. Often it consists of calculating the outcome of some numerical problem.

Expert systems, on the other hand, mostly try to formulate some useful advice to handle problems, or to determine a hypothesis to account for the observed facts. In order to do so, expert systems predominantly use symbolic reasoning. For instance, the problem of determining a probable structural formula of an organic compound from its mass spectrum cannot be solved as a numerical problem. The line of reasoning of an expert system is not completely fixed. In fact, it is the type of problem and the state of the reasoning process which determine the next instruction to be evaluated by the computer. The system selects an instruction which can be carried out, e.g. to establish a part of the molecular formula that is not yet known.

Many expert systems are consultation programs, and therefore, communication between system and user must be possible. The system asks for information about the problem. An important aspect is that the system is flexible with respect to the dialogue, allowing different levels of communication. For a user who is familiar with the system short questions and short answers are most suitable. A user who is not familiar with it, however, needs additional explanations. The system must be able to explain the meaning of a question and the reason why it is posed. After solving the problem, the system should give evidence for its solution. The user will only accept the solution if he understands it and knows how the system has arrived at it. Many systems, such as MYCIN, can explain the meaning of their questions, and can give evidence for their solution by rendering the followed line of reasoning.

Although they can be employed, conventional computer languages such as FORTRAN and PASCAL are not very suitable for symbolic reasoning. Usually, more suitable languages are used in expert systems, especially LISP [6] and PROLOG [7]. In LISP, for instance, there is essentially no difference between statements and data. Because of this, the knowledge containing part of an expert system can be separated from the part of the system controlling the problem solving process, which greatly improves the system's flexibility.

Knowledge representation

Expert systems must be able to deal with knowledge in an abstract manner. Generally speaking, knowledge in a domain consists of descriptions that characterize objects and relationships, and procedures for manipulating these descriptions. In different domains the knowledge can take different forms. A form frequently encountered consists of empirical associations: certain observed facts, for instance, give an indication of the existence of some phenomenon. In other domains causal models may apply.

The knowledge can be obtained from textbooks, but also from the experience of human experts. Especially the latter type of knowledge is in many cases theoretically not well founded and has only a limited applicability. However, it renders the flexibility to efficiently handle real-world problems. All chunks of knowledge must be made available to the expert system by translation into a suitable knowledge representation.

There are several ways to represent knowledge. Many expert systems employ knowledge representations that are based on predicate calculus, a formal language of symbol structures. Simple declarative facts can be represented as predicates. For instance 'Copper is an element' and 'Colour of copper is yellow' can be represented as `ELEMENT(copper)` and `COLOUR(copper,yellow)`, respectively. These statements can be combined by means of logical operators, e.g.

`ELEMENT(copper) ^ COLOUR(copper,yellow)`

to denote that copper is an element and has a yellow colour. Predicate calculus also allows procedural representation of knowledge. The statement 'The phase of copper is solid at room temperature' could be expressed as:

`(∀x) [COPPER(x) ^ TEMPERATURE(x,roomtemp.) -->
--> PHASE(x,solid)]`

The actual representation in expert systems frequently deviates from the traditional notation. Particularly, systems written in LISP employ list structure notation. Both statements could become then:

`(AND (ELEMENT copper) (COLOUR copper yellow))`

`(ALL (x) (IF (AND (COPPER x) (TEMPERATURE x roomtemp.))
(PHASE x solid))))`

The second type of statement is a rule which more conveniently arranged would read:

IF the substance x is copper

AND the temperature of x is room temperature

THEN the phase of x is solid.

The knowledge of many expert systems is largely encoded in these IF ... THEN ... rules. These systems are generally called rule-based production systems, in which production stands for logical inference. The rules themselves are referred to as production rules.

Figure 1. Example of a semantic or relational network. It contains some properties of copper hydroxide and its constituents.

A second way to represent knowledge is by means of a relational or semantic network. The objects to be described are represented as nodes in a graph, and the relationships between objects are represented as links from one node to another. For every link also the type of relationship is recorded. Figure 1 contains a semantic network with some properties of copper hydroxide and its constituent elements. This type of representation is very suitable for storing declarative knowledge. Because an object is linked to related objects, searching the database can be very efficient. However, introduction of a new object into a large semantic network will be difficult, and procedural knowledge can hardly be represented.

Also frame-like structures can be used for knowledge representations. A frame is a data structure that contains all relevant knowledge about an object stored together. Several variants of frame representation are possible, but most employ different types of frames for different types of objects. A frame, then, has slots or fields where facts can be filled in. These facts can refer to other frames. In figure 2, for instance, one of the constituents of COPPER HYDROXIDE is HYDROGEN, which refers to another frame. This type of representation allows inheritance from one object (frame) to another. As is stated in figure 2, HYDROGEN is a specialization of ELEMENT which implies that all facts about ELEMENT are inherited by HYDROGEN. This inheritance hierarchy reduces the number of slots to be filled in because the properties of a class of objects gathered in one frame can be used by the frames of the specializations of this class.

Most expert systems use production rules for knowledge representation. MYCIN's knowledge base consists of some 500 rules. Many systems use more types of knowledge representation. For example, CENTAUR [8], belonging to the offspring of MYCIN and also a medical diagnosis program, makes use of production rules but it also has a frame-like structure. Every frame corresponds to a disease in the domain, and contains a set of rules and characteristics of the disease, e.g. the range of plausible results of a medical test. The system reasons from one frame to another. At any time, only one frame is active and only the rules and facts that are listed in it may be used. Because of its

Figure 2: Example of a frame representation. The two frames contain knowledge about copper hydroxide and hydrogen.

frame-like structure, the knowledge base gains in clarity and a search for an applicable rule is less extensive.

The knowledge base of the medical diagnosis program CASNET [9,10] is structured as a Causal ASSociational NETWORK. The network contains nodes that represent patho-physiological stages. These nodes can be connected by causal links. By reasoning along the causal links from one node to another the system tries to determine the severity of the disease which can be derived from the status of the nodes. Connected to each node is a rule which is applied to establish its status: it can be denied, confirmed or unknown. Generally speaking, the more nodes are confirmed the more severe will be the disease.

Manipulation of knowledge

In order for a system to reason, it must be able to manipulate the data of a specific problem using the general facts of the domain (declarative knowledge) and the rules (procedural knowledge) until a satisfying solution is found or no more rules can be applied. The control of the reasoning process is strongly dependent of the type of domain and knowledge representation. In fact, something about control has already been mentioned in the previous section. However, it concerns only specific aspects which are not generally applied in production systems.

A more general aspect of production systems is that the reasoning process consists of a recognize-act-cycle. This cycle consists of: find all applicable rules, select one, apply it and carry out its conclusion (then-part). The cycle repeated until a goal is reached or no more rules are applicable. It is the control of the system which directs this cycle. A production system consists of three parts: a global database with declarative knowledge, a production or rule base with the procedural knowledge and a control, as is schematically depicted in figure 3. The control part is often referred to as inference engine.

Figure 3: Structure of a production system

With respect to the control, two major approaches are possible. The first approach is forward chaining. The reasoning starts from an initial set of facts (the existing global data base). A rule is selected whose precondition (if-part) matches the facts in the data base. This rule is applied resulting in one or more changes in the data base, e.g. the addition of the inferred fact. The cycle is repeated until the goal state is reached. This type of reasoning is called data or event driven. However, if the desired goal state can be formulated beforehand and there are many possible initial states, another approach will be more efficient: backward chaining. In this case, the reasoning starts from the goal state. A rule is applied which results in a subgoal. The cycle is repeated until the subgoals match the facts in the data base. Backward chaining is called goal or expectation driven. A thorough discussion of this subject has been given by Nilsson [11].

Figure 4: Example of a simple AND/OR graph for a backward chaining system. The AND-branches are denoted by arcs.

Resolving a problem by an expert system can be represented as searching a path in an AND/OR graph leading from the initial node to ending nodes. Figure 4 is an example of a simple AND/OR graph for a backward chaining system. Node 1 is the initial node; it is the goal state. It is true if node 2 and node 3 are true, or if node 4 is true. In fact, this is the graphical representation of a rule. Node 2, in turn, can be confirmed from the ending nodes 5 and 6. If node 5 cannot be confirmed from the facts in the data base this path is inconsistent, and the reasoning process must return to node 2. This is called backtracking. If also node 6 cannot be confirmed, the system must backtrack to node 1 and inspect the path leading through node 4.

MYCIN, for instance, employs backward chaining. Every consultation starts with the following (simplified) rule:

IF there is an organism which requires therapy
THEN determine the best therapy
ELSE indicate that the patient does not require therapy.

In order to apply this rule, the system must know whether there is an organism. One of MYCIN's rules tells that organisms associated with significant disease require therapy. The system does not know anything about organisms yet. Therefore, it asks the user if there are any organisms. To apply the rule, however, it must also know whether the organism is significant. Now other rules can be applied to make a decision about this. This way of reasoning is continued until no piece of relevant information is missing anymore, or no more rules can be applied. Usually, the system tries to infer a piece of missing information first, before it asks the user.

The level of control described thusfar suffices only for simple domains with reliable and static data and knowledge, and a small space of possible solutions. Most domains fail to meet these requirements [12]. Also MYCIN's control structure is actually more complex than has been described here. In many domains the space of possible solutions is too large. It frequently occurs that more rules are applicable at the same time. It takes too much time to simply take one and try another if the

first one fails to be a step in the right direction. Exhaustive search must be avoided. In order to improve the efficiency of the search strategy, additional control is necessary. Usually, this control makes use of the specific properties of the domain. Sometimes a rule can be used that tells that some type of rule is more likely to be successful than another. In fact, this rule selects the rule to be applied in the reasoning process. It acts on other rules, and therefore, it is called a meta-rule. It is also referred to as a heuristic because it affects the search strategy.

Many domains allow specific methods to avoid exhaustive search. For instance, DENDRAL's task is to derive all molecular structures of a compound that are consistent with its mass spectrum. The basic approach is to consider all possible structures and to rule out those inconsistent with the spectrum. This would imply that for larger molecules millions of structures should be considered. However, the program already has some knowledge about the problem: a set of constraints, e.g. about the presence of functional groups. These constraints may have been derived from a preliminary inspection of the spectrum by the program, or may have been entered by the user. DENDRAL's structure generating program GENOA [13] generates only structures that are in accordance with the constraints. Only these structures need further inspection which greatly reduces the search effort.

Dealing with inexact knowledge

In many domains an expert system must be able to deal with uncertainty in the experimental data or in the rules. Sometimes the redundancy of the data can be used to correct wrong data. This can only be accomplished if the system is enabled by meta-rules to evaluate the reliability of the data based on their origin. This method of dealing with uncertainty is employed by GA1 a system for structural analysis of DNA molecules [14].

Mostly, however, it is impossible to overrule wrong data, and then, a mechanism is needed for propagation of uncertainty. Also many rules

contain uncertainty because they are derived from a specialist's experience and will not always apply. Even if the facts in the precondition of the rule are absolutely certain, application of such a rule results in a conclusion that is not absolutely certain. For several systems mechanisms have been developed to deal in a numerical way with uncertainty in facts and rules [15,16], e.g. the medical diagnosis programs CASNET and MYCIN and the consultation program for mineral exploration PROSPECTOR [17,18]. Only the confirmation theory used in MYCIN is briefly considered here.

Each fact and each production rule are associated with a certainty factor CF, a number in the interval $[-1,1]$. The CF indicates the certainty with which a fact or rule is believed. Negative and positive values indicate a predominance of opposing and confirming evidence, respectively. The values -1 and 1 indicate the absence of uncertainty. Experience with human experts shows that they do not use information in a way compatible with standard statistical methods. Therefore, CFs which are not probabilities are applied instead of standard statistical methods. Consider, for instance, the following production rule of MYCIN:

IF (1) the infection is primary-bacteremia, and
(2) the site of the culture is one of the sterilesites, and
(3) the suspected portal of entry of the organism is the gastrointestinal tract,

THEN there is suggestive evidence (0.7) that the identity of the organism is bacteroides.

The CF of this rule is 0.7, i.e. the CF of the conclusion is 0.7 if the precondition is absolutely certain. Frequently, however, one or more facts of the precondition do not represent absolute knowledge. To evaluate such a production rule, one applies the following steps:

- the CF of a conjunction of several facts is taken to be the minimum of the CFs of the individual facts;
- the CF of the conclusion of a rule is the arithmetic product of the CFs of the precondition and the rule itself;

- the CF of a fact which is the conclusion of two or more rules is taken to be the maximum of the CFs of the individual conclusions of the rules.

For example, if the second fact of the rule mentioned above has been assigned $CF = 0.8$ to, and both other facts have CFs of 0.95 and 1.0, respectively, then the CF of the conclusion of the rule will be 0.56. The CFs of individual rules are estimated by the specialists whose expertise the rules are based on.

Expert systems try to model the reasoning process of human experts. They make use of knowledge which is derived from human expertise, and therefore, judgemental, experiential, and uncertain. Mechanisms to deal in a numerical way with uncertainties have been introduced to emulate the expert's ability of inexact reasoning. However, it has been shown [19] that human subjects are more effective in reasoning with verbal expressions than with numerical expressions. They preferably employ statements such as:

IF an analysis is used for process control
THEN use a fast analytical procedure.

A fast analytical procedure is usually expensive.

Fuzzy set theory provides a method to deal with these verbal expressions [20].

In classical set theory an element either belongs to a set or not, there is no intermediate. It is governed by a logic that permits a proposition to be either true or false. This logic is not suitable for representing vague concepts such as fast, expensive, and so on. The boundary between fast and not fast is too distinct in this logic. This may result in a situation that an analytical procedure requiring 15 minutes to be carried out is called fast, while another requiring 16 minutes is called not fast. Fuzzy sets, on the other hand, allow a degree of vagueness. An element may belong only partly to a fuzzy set, it has a degree of membership. This degree is a number in the interval $[0,1]$. A proposition need not be simply true or false but may be partly true. Consider, for instance, the ana-

Table 1

Several analytical procedures for nitrogen determination and their degree of membership to the set 'Fast analytical procedures'.

Analytical procedure	Analysis time (min.)	Degree of membership
Total N, classical distillation	75	0.0
Total N, automated analyzer	12	0.4
NO ₃ -N, auto-analyzer	15.5	0.3
NO ₃ -N, ion selective electrode	10	0.6
NH ₄ NO ₃ /CaCO ₃ ratio, x-ray diffr.	8	0.8
Total N, neutron activation	5	0.9
Total N, gamma-ray absorption	1	1.0

lytical procedures for nitrogen determination [21] in table 1. According to their analysis times degrees of membership to the set fast analytical procedures are assigned to the procedures. The proposition 'The procedure using ion selective electrodes is a fast analytical procedure' is only partly (0.6) true. In the reasoning process ion selective electrodes may still be considered when a fast procedure is needed, although other have a higher score at this point. Several operators are defined for manipulation of fuzzy sets, e.g. conjunction and disjunction of two fuzzy sets. A production rule becomes a relation between two fuzzy sets. Fuzzy set theory has not been frequently applied in expert systems yet. However, this approach is gaining importance [20,22-24].

TOOLS FOR BUILDING EXPERT SYSTEMS

Building an expert system is a time consuming and difficult process which is certainly not yet standardized. Two major stages can be discerned in this process: knowledge acquisition and the development of a control structure of the system. Knowledge acquisition comprises extracting knowledge from human experts and textbooks, and encoding it in a suitable representation. This implies that the bulk of puzzling complexity containing the facts, theoretically founded rules and rules of thumb the expert uses for problem solving, must be ordered into a set of non-conflicting rules and facts which can be used by the expert system. Even the specialist is not able to state in a straightforward manner the chunks of knowledge he uses. They must be extracted one by one by asking the specialist all the time what has to be done in a specific situation. Automated knowledge acquisition in a self-adaptive, learning expert system would greatly reduce the development time. Something has already been accomplished in this field, e.g. meta-DENDRAL which will be considered in detail later. In general, however, it must be admitted that research in this field is still in an early stage.

It is important that the control part or inference engine of an expert system is separated from the knowledge containing part. Especially in the development phase the possibility of correcting wrong rules, for instance, must exist while the other parts of the system remain unchanged. The development of an inference engine with a sufficient degree of generality is difficult and error prone. However, the separation of knowledge and the more or less general inference engine opens new possibilities to simplify the building of an expert system. Replacement of all domain-specific knowledge by knowledge of another domain results in a new expert system. An expert system from which all domain-specific knowledge has been removed is called empty shell.

MYCIN's domain, for instance, is the diagnosis of infectious blood diseases. From this system a domain independent version was produced called EMYCIN [25]. This system contains all of MYCIN except its knowledge of infectious blood diseases. EMYCIN formed the basis for another medical diagnosis program PUFF for pulmonary diseases [26],

which is operational in the Pacific Medical Center in San Francisco. EMYCIN has also been used for building expert systems that are not medical diagnosis programs, e.g. LITHO [27,28], a consultation program for underground geology.

Several other empty shells are derived from domain-specific expert systems. KAS is derived from PROSPECTOR, EXPERT [29] from CASNET, and HEARSAY-III [30] from the system for speech-understanding HEARSAY-II [31]. Apart from HEARSAY-III which employs a much more sophisticated problem solving strategy, these empty shells can only be used for building consultation programs. They employ the same reasoning strategy as the systems they are derived from. EMYCIN, for instance, can only tackle diagnosis problems where backward chaining is a suitable strategy. These empty shells (except HEARSAY-III) provide not only an inference engine but also a knowledge acquisition facility permitting rapid construction of a knowledge base, and a consultation facility for a smooth interaction between user and system and for explaining the followed line of reasoning.

More general with respect to problem solving strategy and knowledge representation are rule-based programming systems such as OPS5 [32] and ROSIE [33]. One of the most well-known expert systems, R1 for configuring computer systems [34], is written in OPS5. The increased generality makes these programming systems applicable in a broader range of problems. However, building an expert system is more time consuming because they lack sophisticated facilities for knowledge base construction, and OPS5 lacks facilities for explanation and for user-system interfacing.

Application of these tools speeds up the developing process, and the knowledge acquisition can be fully concentrated on. It is important to check beforehand whether the domain meets the demands of the empty shell. In [35] several empty shells are used to build an expert system for the same domain in a limited time. This comparison shows fairly well the properties and the limitations of empty shells.

APPLICATION OF EXPERT SYSTEMS IN CHEMISTRY

Chemistry has always been one of the major application fields of expert systems. A few general articles on expert systems have already been published in the chemical literature [38-40]. With respect to expert systems in chemistry three areas of concentrated research can be discerned, one of which belongs to analytical chemistry. It concerns chemical engineering, organic synthesis planning and structure elucidation.

Table 2

Synthesis planning programs. The numbers refer to the references

LHASA		42-45
SECS		46
SYNCHEM		47-48
Bersohn's program		49
EROS		50-54
AHMOS		55

The task of synthesis planning programs is to determine the synthesis pathway for a particular organic compound. Most programs reason in the retro-direction, i.e. precursors of the goal molecule are determined. This process is repeated until a satisfying set of starting chemicals for the synthesis is determined. This process can be represented as an AND/OR graph which is searched breadth-first or depth-first. To find the precursors, a transform library of known reactions is used. This group of programs has been reviewed by Haggin [41]. In table 2 several synthesis planning programs are listed with the most recent references. Some programs are used by the chemical industry, e.g. LHASA by Dupont and SECS by Merck, Sharp & Dohme.

Currently, there is an increasing interest in application of expert systems in chemical engineering. An expert system has been reported to calculate the risk of various factors involved in stress-corrosion-cracking in stainless steel (SCCES) [56]. It is a consultation program with a relatively simple MYCIN-like structure based on a commercially available inference engine. Its knowledge base essentially is a causal model which was constructed by two corrosion specialists and one computer specialist in approximately six man-weeks. Another application in chemical engineering is HEATEX [57], a system for aiding in the construction of networks that minimize energy requirements by allowing the exchange of heat among various process streams. It is written in one of the predecessor languages of OPS5. Related is the expert system CONPHYDE [57,58] which is designed to aid an engineer in the selection of an appropriate vapour-liquid equilibrium method when performing various process calculations. The system is based on the empty shell KAS.

These systems have in common that they are relatively simple consultation programs which have been developed because of the scarcity of available experts. Expert systems are always available and can easily be reproduced into several copies. Another promising application field arises from the increasing complexity of chemical process plants, which implies more data for the operator to attend to and increased chances for human error. FALCON [59] is an expert system to efficiently handle many real-time data for alarm analysis in chemical process plants. While it is being developed real-time data are provided by a simulation model of the process.

The problem of having too many data to attend to also exists in analytical chemistry. A triple quadrupole mass spectrometer (TQMS) is a very complex instrument requiring selective tuning or optimization of over 30 parameters. The problem is how to optimize for the most significant data as opposed to collecting the most data. Therefore, the expert system TQMSTUNE [60-62] is being developed. Its most important aspect is its ability to do on-line real-time optimization of the data selection, acquisition, and interpretation. A related application of an expert system is VM [63] which monitors mechanical ventilation of intensive-care patients.

Table 3

Systems for structure elucidation and the types of data that can be used. The numbers refer to the references.

	MS	¹ H NMR	¹³ C NMR	IR	UV	References
DENDRAL	y		y			3 13 36 37
STREC	y	y		y	y	64 65
Damo	y					66
CHEMICS		y	y	y		67-69
SEAC		y		y	y	70 71
SCANSPEC	y	y	y	y	y	72

Beside DENDRAL, several other systems for structure elucidation have been or are being developed. Basically, most systems employ the same problem solving strategy as DENDRAL's:

- inference of a set of constraints from an inspection of the spectrum; mostly, it concerns functional groups;
- generation of all possible structures according to the constraints, and prediction of their spectra (model driven approach);
- comparison of predicted and real spectrum in order to prune unsatisfying structures.

Most systems consist of several modules for different spectroscopic methods. In table 3 several systems are listed with their spectroscopic methods and references. Apart from DENDRAL, these systems are written in conventional languages such as FORTRAN, which can be considered as a serious drawback for this type of problems.

Some systems employ a purely data-driven strategy which consists of relating spectral characteristics to structural characteristics. If, for instance, in IR there is a peak in the range $1765\text{--}1660\text{ cm}^{-1}$, then the presence of an aldehyde group becomes more probable. This strategy is used by Woodruff's IR analysing program [68], and by ISIS [73] a

micro-computer program for structure elucidation using IR and MS spectral data in an integrated manner. DENDRAL's ^{13}C NMR program can employ both the purely data-driven and the more model-driven approach. The former approach has the advantage that it is closer to the approach of human experts. Therefore, it is suitable for a system for educational purposes.

However, there is a serious limitation to the applicability of the purely data-driven approach as it comes to determining all structures that match the spectrum. The spectral characteristics of all functional groups are stored in a data base. A compound containing a functional group that is not present in the data base, will never be properly analysed. In case of a large data base, it will be difficult to detect the omission. A program employing the other approach also uses a data base with spectral characteristics to predict the spectrum of a generated structure. Neither this program will be able to correct for missing information in the data base. The structure with the missing functional group will be generated, and the omission will be easily detected.

According to Hippe [72], the systems for structure elucidation operate at the level of a post doc analyst. This can mainly be attributed to their systematical application of the whole set of rules each time, without mistakes and loss of memory.

A related domain is covered by the system CRYNALIS (formerly SU/P) [74,75]. Its task is to derive three dimensional models of crystallized proteins from X-ray diffraction data. Its control structure is related to HEARSAY-II's. There are several control layers, the highest of which is the strategy layer. In this layer it is determined what work has to be done to solve the problem, and the inference work is done in layers lower in the hierarchy.

DISCUSSION AND FUTURE EXPECTATIONS

An important factor in the success of expert systems is the adaptation of their control structure to the specific properties of the domain. The control structure cannot be completely general because it must be able to handle the knowledge to avoid exhaustive search. Frequently, however, a similar control structure can be employed in various domains which may be as different as underground geology and medicine (the development of LITHO with use of EMYCIN). If a new domain can be modeled according to the requirements of an empty shell a great reduction in the development time of the new expert system can be achieved. A careful examination of both the new domain and the empty shell is of the utmost importance, in order to avoid a situation that an empty shell is bought which appears not to be suitable for the given domain.

In analytical chemistry there are many applications for expert systems which are relatively simple and which allow the use of an empty shell. For instance, in many laboratories only a limited number of different sample types are processed. The knowledge relevant to these types can be gathered into an expert system. Apart from expert advice, there are also other important roles for this type of expert systems. It can be used as a checklist because the expert system will reliably pose all relevant questions. It can be used for training of novices or for refining the expertise of the specialists. A very important role for the system to play is to combine the advice with the administration of entering samples and the produced analytical results, and to plan the sample routing.

Most empty shells prefer domains that are deep and narrow, i.e. the line of reasoning consists of several steps and at any stage only few rules are applicable. In analytical chemistry many domains are broad and shallow. Consider, for instance, an expert system covering the domain of AAS. It will only be relevant if it is able to do more than telling the user which lamp is suitable for a particular element. It must be able to give advice in sample preparation, and in problems that may occur during the analysis, e.g. matrix effects. This implies that the system must be able to process characteristics of the sample itself. As a result, the domain becomes rather broad. This type of application

requires a specialized control structure, in order to avoid that too many questions must be answered by the user. This would be a severe limitation to the applicability in practice.

Expert systems for practical advice concerning analysis by AAS, or even concerning analytical chemistry in general, must have knowledge about analytical procedures. Up to now this knowledge is stored in the literature and mainly consists of descriptions of procedures for the analysis of a compound in a particular matrix. Efficient use of the knowledge requires refinement, e.g. removal of redundancies, and a suitable representation. Application of expert systems will be likely to cause a renewed interest in the issue of systematization which is one of the most important - and one of the most neglected - issues of analytical chemistry. Which are the factors that determine the quality of an analytical procedure? How can an analytical procedure be described in an informative way?

Figure 5: Example of SSSAC. Symbolic representation of the procedure for the complexometric determination of ferric iron in Renn slag, with potentiometric indication (from Malissa et al. [76]).

In this respect it is interesting to consider Malissa's proposals for the introduction of a symbolic iconic language in analytical chemistry SSSAC (Symbolic Synoptic System for Analytical Chemistry) [76-78].

SSSAC consists of informative symbols for the description of analytical devices and procedures and their characteristics (figure 5) It appeared not to be able to force a major break-through, probably because of the difficulty at that time to process the proposed iconographic symbols by computer. Currently, however, it is the AI research that has shown new directions for the processing of symbols by computer. A mixed representation approach seems to be most promising. In this approach the analytical symbols are internally represented in alphanumerical code, which can be processed more easily by the computer. This code can be translated into Malissa's informative iconographic symbols for external use.

An interesting application field of expert systems is control of complex analytical instruments such as a triple quadrupole mass spectrometer. Expert systems can serve as an alternative for instrument optimization procedures such as the simplex and the steepest ascent method. These are usually rather successful, but they need a large number of measurements, especially when there are many interacting parameters. They are not able to make use of the characteristics of the instrument. Expert systems, on the other hand, can, which will likely result in fewer measurements needed for the optimization. Flexible control of the instrument and the subsequent data analysis requires a specialized control structure of the expert system with several search spaces for the different modes of the instrument.

For applications which are more complex construction of a dedicated and specialized control structure is needed because it can make use of the properties of the domain. Development of a general AI language will be of great help because the applicability of empty shells is limited to an area of related domains, and LISP itself is not a very user-friendly programming environment.

It is only recent that expert systems are becoming more widely known. Many essential questions are not yet answered. For instance, will expert systems be accepted by the practical workers in the field? In order to answer this question, there must be more experience with the application of expert systems in everyday practice. Indeed, research on many systems that have been discussed here, is rather concerned with

the development of AI techniques than with application in practice. It is important that expert systems become widely known, also in chemistry. The research on expert systems will benefit from relevant problems handed over by the practical chemists, and from application of expert systems in practice. While accomplishing their routine tasks, the practical chemists, on the other hand, will benefit from a powerful and flexible support expert systems are able to render.

REFERENCES

- 1 A Newell et al, GPS a program that simulates human thought, in E A Feigenbaum, J A Feldman (eds), Computers and Thought, McGraw-Hill, New York, 1963
- 2 E H Shortliffe, Computer-based medical consultation MYCIN, Elsevier, New York, Oxford, Amsterdam, 1976
- 3 R K Lindsay, B G Buchanan, E A Feigenbaum, J Lederberg, Applications of Artificial Intelligence for Organic Chemistry - The Dendral Project, McGraw-Hill, New York, 1980
- 4 R E Shannon, R Mayer, H H Adelsberger, Simulation, 44(1985)275
- 5 J V Hansen, W F Messier, Int J Comput & Inf Sci, 11(1982)357
- 6 R Wilensky, LISPcraft, Norton & Co, New York, London, 1984
- 7 K L Clark, F J McCabe, Micro-PROLOG programming in Logic, Prentice/Hall, Englewood Cliffs NJ, 1984
- 8 J S Aikins, Artif Intell, 20 (1983) 163
- 9 S M Weiss, C A Kulikowski, S Amarel, A Safir, Artif Intell, 11 (1978) 145
- 10 S Weiss, A Safir, Computers in Biology and Medicine, 8 (1978) 25
- 11 N J Nilsson, Principles of Artificial Intelligence, Tioga, Palo Alto, 1980
- 12 M Stefik, J Aikins, R Balzer, J Benoit, L Birnbaum, F Hayes-Roth, E Sacerdoti, Artif Intell, 18 (1982) 135
- 13 R E Carhart, D H Smith, N A B Gray, J G Course, C Djerassi, J Org Chem, 46 (1981) 1708
- 14 M Stefik, Artif Intell, 11 (1978) 85
- 15 P S Szolovits, Artif Intell, 11 (1978) 145
- 16 R Martin-Clouaire, H Prade Int J Man-Machine Studies, 22 (1985) 251
- 17 R O Duda, P E Hart, P Barrett, J Gaschnig, K Konolige, R Reboh, J Slocum, Model design in the PROSPECTOR consultant system for mineral exploration, in D Michie (ed), Expert systems in the micro-electronic age, Edinburgh, 1979

- 18 R O Duda, P F Hart, N J Nilsson, G Sutherland, Semantic network representation in rule-based inference systems, in D A Waterman, F Hayes-Roth (eds), Pattern Directed Inference Systems, Ac Press, New York, 1978
- 19 A F Zimmer, Verbal vs numerical processing of subjective probabilities, in R W Scholz (ed), Decision making under Uncertainty, Elsevier, Amsterdam, 1983
- 20 C V Negotta, Expert Systems and Fuzzy Systems, Benjamin/Cummings, Menlo Park Ca , 1985
- 21 G Kateman, F W Pijpers, Quality Control in Analytical Chemistry, Wiley, New York, 1981
- 22 L Lesimo, L Saitta, P Torasso, Int J Man-Machine Studies, 22 (1985) 307
- 23 W Bandler, L J Kohout, Int J Man-Machine Studies, 22 (1985) 347
- 24 W van Melle, A domain-independent production-rule system for consultation programs, Proceedings of the Sixth International Joint Conference on Artificial Intelligence, 1979 p 923
- 25 J C Kunz, R J Fallat, D H McClung, J J Osborn, R A Votteri, H P Nu, J S Atkins, L M Fagan, E A Feigenbaum, A physiological rule-based system for interpreting pulmonary function test results, Working paper HPP-78-19, Heuristic Programming Project, Dept of Computer Science, Stanford University, 1978
- 26 M Dohnal, Comput Ind , 6 (1985) 115
- 27 A Bonnet, J Harry, J-G Ganascia, Tech & Sci Inf , 1 (1982) 393
- 28 J-G Ganascia, Int J Man-Machine Studies, 20 (1984) 229
- 29 S Weiss, C A Kulikowski, EXPERT A system for developing consultation models, Proceedings of the Sixth International Joint Conference on Artificial Intelligence, 1979, p 942
- 30 R Balzer, L D Erman, P London, C Williams, HEARSAY-III A domain-independent framework for expert systems, Proceedings of the First National Conference of the American Association for Artificial Intelligence, 1980, p 108
- 31 L D Erman, F Hayes-Roth, V Lesser, D Reddy, A C M Comput Surveys, 12 (1980) 213
- 32 C L Forgy, The OPS5 user's manual, Technical Rept , CMU-CS-81-135, Computer Science Department, Carnegie-Mellon University, Pittsburgh Pa , 1981
- 33 J Fain, D Gorlin, F Hayes-Roth, S J Rosenschein, H Sowizral, D Waterman, The ROSIE language reference manual, Tech Rept N-1647 ARPA, Rand Corp , Santa Monica Calif , 1981
- 34 J McDermott, Artif Intell , 19 (1982) 39
- 35 F Hayes-Roth, D A Waterman, D B Lenat (eds), Building expert systems, Addison-Wesley, Reading Mass , 1983
- 36 N A B Gray C W Crandell, J G Nourse, D H Smith, M L Dageforde, C Djerassi, J Org Chem , 46 (1981) 703
- 37 N A B Gray, Artif Intell , 22 (1984) 1
- 38 R E Dessy (ed), Anal Chem, 56 (1984) 1200A
- 39 J Fox, Trends Anal Chem , 4 (1985) IX

- 40 J W A Klaessens, G Kateman, B G M Vandeginste, Trends Anal Chem , 4 (1985) 114
- 41 J Haggin, Chem Eng News, 61 (1983) 7
- 42 E J Corey, N Frank Feiner, J Org Chem , 45 (1980) 757 & 765
- 43 E J Corey, A P Johnson, A K Long, J Org Chem , 45 (1980) 2051
- 44 E J Corey, A P Johnson, A K Long, J Chem Inf Comput Sci , 20 (1980) 221
- 45 A K Long, S D Rubenstein, L J Joncas, Chem Eng News, 61 (1983) 22
- 46 P Gund, J D Andose, J B Rhodes, G M Smuth, Science, 208 (1980) 1425
- 47 K K Agarwal, D L Larsen, H L Gelemier, Comp Chem , 2 (1978) 75
- 48 H Gelemier, G A Miller, D L Larsen, D J Berndt, Realization of a large expert problem-solving system - SYNCHEM2 a case study, First Conference on Artificial Intelligence Applications Denver Co , IEEE, Silver Spring MD , 1984, p 92
- 49 M Bersohn, K Mackay, J Chem Inf Comput Sci , 19 (1979) 137
- 50 I Ugi, J Bauer, J Brandt, J Friedrich, J Gasteiger, C Jochum, W Schubert, Angew Chemie, 91 (1979) 99
- 51 C, Jochum, J Gasteiger, I Ugi, Z Naturforsch , 37b (1982) 1205
- 52 M D Guillen, J Gasteiger, Tetrahedron, 6 (1983) 1331
- 53 J Gasteiger, M Marsili, Org Magn Res , 15 (1981) 353
- 54 J Gasteiger, M Marsili, Tetrahedron, 36 (1980) 3219
- 55 J H Winter, Chemische Syntheseplanung in Forschung und Industrie, Springer, Berlin, Heidelberg, New York, 1982
- 56 A Basden, Int J Man-Machine Studies, 19 (1983) 461
- 57 S R Reddy, N Tyle, M L Maher, R Banares, M D Rychener, S J Fenves, Knowledge-based expert systems for engineering applications, 1983 Proceedings of the International Conference on Systems, Man and Cybernetics, Bombay and New Delhu, India, IEEE, New York, 1984, p 364
- 58 R Bañares-Alcántara, A W Westerberg, Comput & Chem Eng , 9 (1985) 127
- 59 D Chester, D Lamb, P Dhurjati, Rule-based computer alarm analysis in chemical process plants, Proceedings of the Seventh Annual MICRO-DELCON '84, The Delaware Bay Computer Conference, Newark, IEEE, Silver Spring MD , 1984, p 22
- 60 C M Wong, R W Crawford, J C Kunz, T P Kehler, IEEE Trans Nucl Science, NS-31 (1984) 804
- 61 C M Wong, R W Crawford, Int J Mass Spectrom & Ion Process , 60 (1984) 107
- 62 R E Dessy (ed), Anal Chem , 56 (1984) 1312A
- 63 J J Osborn, L M Fagan, R J Fallat, J C Kunz, D H McClung, R R Mitchell, Medical Instrumentation, 13 (1979) 330
- 64 V V Serov, M E Elyashberg, L A Gribov, J Mol Struct , 31 (1976) 381
- 65 L A Gribov, Anal Chim Acta, 122 (1980) 249
- 66 C Damo, C Dachun, K Teshu, C Shaoyu, Anal Chim Acta, 133 (1981) 575
- 67 S Sasaki, I Fujiwara, H Abe, T Yamasaki, Anal Chim Acta, 122 (1980) 87

- 68 I Fujiwara, T Okuyama, T Yamasaki, H Abe, S Sasaki, Anal Chim Acta, 133 (1981) 527
- 69 H Abe, T Yamasaki, I Fujiwara, S Sasaki, Anal Chim Acta, 133 (1981) 499
- 70 B Debska, J Duliban, B Gusowska Swider, Z Hippe, Anal Chim Acta, 133 (1981) 303
- 71 Z Hippe, Anal Chim Acta, 150 (1983) 11
- 72 Z Hippe, J Chem Inf Comput Sci, 25 (1985) 344
- 73 H J Lunge, H.A van 't Klooster, Trends Anal Chem, 4 (1985) 242
- 74 E A Feigenbaum, Themes and case studies of knowledge engineering, in D Michie (ed), Expert systems in the micro-electronic age, Edinburgh, 1979, p 1
- 75 H P Nii, E A Feigenbaum, Rule based understanding of signals, in D A Waterman F Hayes Roth (eds), Pattern directed inference systems, Ac Press, New York 1978, p 483
- 76 H Malissa, G Jellinek, Fres Z Anal Chem, 247 (1969) 1
- 77 H Malissa, V Simeonov, Fres Z Anal Chem, 289 (1978) 257
- 78 G Kateman, Fres Z Anal Chem, 293 (1978) 388

Part 3

DEVELOPMENT OF SOFTWARE MODULES

ANALYSIS OF LABORATORY SAMPLE INPUT SIGNALS*

SUMMARY

An expert system is presented for automated time series analysis of laboratory sample input signals. The system, AUTOCORR, builds a model of the time series by identifying the processes that are present. These are an uncorrelated random process and, underlying, possibly one or more of the following: a first order autoregressive process; a trend, and a periodic process. AUTOCORR has a knowledge base of 44 rules and 41 facts for this purpose. The employed shell, INFER, allows the use of algorithmic procedures. Elaborate tests with simulated signals show that AUTOCORR has a very low false positive score and is successful in describing time series for laboratory simulation models.

INTRODUCTION

Simulation of laboratory sample input signals is an important part of the simulation model of an analytical laboratory. Several reports are published showing that digital simulation is a useful method for decision support in analytical laboratories [1-4]. These reports also showed that the method was not yet suitable for application in a routine way. In order to increase the cost-effectiveness of the method, and thus the applicability, two software tools have been developed which allow a simulation project to be carried out by the laboratory staff itself.

* This chapter is published as: P. Mijland, J. Klaessens, B. Vandeginste, G. Kateman, Expert system for the characterization of laboratory sample input signals, J. Chemometrics, in press

One of the tools is an expert system LABGEN for automatic construction of a simulation model of the laboratory [5,6] An application to a laboratory for quality control is reported [7] The second tool is FEA (Front End Analysis) for analysis of historical laboratory data in order to obtain the data that are necessary for the construction of a simulation model [8] FEA consists of several programs in order to perform two major tasks The first task is the classification of all sample entries into a limited number of sample types with distinct sets of properties The second task is the characterization of the laboratory sample input signal The program for this task is discussed in the present paper

A simulation model can only be useful if it contains a proper model of the input signal Indeed, one of the major points of interest is how the laboratory responds to the fluctuations of the input A proper characterization cannot be limited to the determination of the mean and the standard deviation of the signal In addition, an analysis for trend, periodicity and autocorrelation effects has to be included (see e.g. [9], chapter 14) Such a time series analysis carried out properly requires experience [10,11] which is not available if the simulation project is carried out by the laboratory staff itself Therefore, this experience has been captured into an expert system AUTOCORR, part of the tool FEA

THEORY

The behaviour of a discrete time series can be described by a combination of stochastic and deterministic processes [12-14] First order approximations of these processes are used to build a suitable model

The stochastic processes are described by a first order autoregressive model, which accounts for any autocorrelation present in the signal The recursive model is given by eq. 1

$$v_t = M + \alpha (v_{t-1} - M) + Z_t \quad (1)$$

$$-1 < \alpha < 1$$

x_t stands for the observed stochastic variable at time t , i.e. the number of samples at day t . x_t is determined by three terms: mean M , an autocorrelation process and an uncorrelated random process Z_t . The autocorrelation process is described by a first order autoregressive model, which is in most cases sufficient [9]. The constraint $-1 < \alpha < 1$ on autocorrelation coefficient α keeps the variance of x_t from becoming infinite. This model can only be attributed to a time series which is stationary, i.e. which has a constant mean and variance. The uncorrelated process Z_t is a normally distributed random process with mean zero and standard deviation $S_{Z,w}$.

Deviations in the mean of the time series are described by deterministic processes, assuming the variance of the uncorrelated process is constant over the whole length of the time series. Two processes are of interest: a trend caused by a long term change in mean and a periodic process caused by a short term change due to the working week of the laboratory and the sample source. Cyclic processes over a longer period (e.g. a month) are assumed not to give a major contribution to the signal.

$$M = M_w + T \cdot t \quad (2)$$

$$w = 1, 2, \dots, P$$

The mean, M , contains a constant M_w and a term modeling a linear trend with slope T (samples/day). If a periodicity is present, each day of the week is modeled separately. A periodicity P therefore is described by P different models (eqs. 1,2). Each model has a unique M_w and $S_{Z,w}$ and a joint T and α . The w indicates which week-day is modeled.

Not all processes are necessarily present in a particular time series. Therefore, in analysing a time series the dominating processes are identified. These are used to build a parsimonious model describing the time series. Then the parameters of this model are estimated. These estimated parameters can be used to simulate new time series with the same statis-

Figure 1. Time series x_t , the number of samples, versus t (days). Time series generated according to eqs. 1 and 2, with overall mean 40 and st. deviation 10 samples. Of the original length of 200 days the first 60 are shown. The generating parameters: a: $M_1 = 40$, $S_{Z,1} = 10$, $\alpha = 0$, $T = 0$, $P = 1$; b: $M_1 = 40$, $S_{Z,1} = 7$, $\alpha = 0.7$, $T = 0$, $P = 1$; c: $M_1 = 47$, $M_2 = 43$, $M_3 = 35$, $M_4 = 35$, $M_5 = 43$, $S_{Z,1} = \dots = S_{Z,P} = 8$, $\alpha = 0$, $T = 0$, $P = 5$.

tical characteristics.

Identification of the processes present in a time series (Figure 1 a,b,c) starts off with the determination of r_k , the estimator for the autocorrelation function, calculated by eq. 3:

$$r_k = \frac{\sum_{i=1}^{N-k} (x_i - \bar{x}) \cdot (x_{i+k} - \bar{x})}{\sum_{i=1}^N (x_i - \bar{x})^2} \quad (3)$$

x_i is the signal at time i , $\bar{x}$ is the mean value of the signal and k is the time lag in the autocorrelogram. This is the major diagnostic tool in the analysis. The estimator r_k has a variance of $1/N$, with N the length of the time series [12]. If more than 95% of all the calculated autocorrelation function values lie within the confidence limits

$$-\frac{1}{N} + \frac{2}{\sqrt{N}} \leq r_k \leq -\frac{1}{N} - \frac{2}{\sqrt{N}} \quad (4)$$

it is concluded that there is no autocorrelation and that the time series is well described by the uncorrelated random process. Otherwise the origin of the correlation has to be determined. A significant value of the autocorrelation function at low lag ($k = 1, 2, 3$) may also be an indication of correlation. Figure 2 shows the correlograms of the time series of Figure 1.

The nature of the autocorrelation is elucidated using the spectral density function or spectrum (Figure 3), which is the Fourier transform of eq. 3. The surface below the spectrum shows the variance distribution over the frequency-range: $I(v) \cdot dv$ is the proportion of variance of the interval $\langle v, v+dv \rangle$. A peak in the spectrum indicates that these frequencies give a major contribution to the time series. A peak at frequencies around $1/P$ (Figure 3c) indicates the presence of a periodicity P . The cosine present in the correlogram also shows periodicity is present in the

Figure 2 Autocorrelograms with autocorrelation value r as a function of lag k , corresponding to the time series of Figure 1 ----- marks the confidence limits (see text) a No significant correlation is present b Correlation present, mainly at low lag c Correlation present, indicating periodicity

Figure 3 Fourier transform of Figure 2, Intensity versus frequency ν , calculated according to Chatfield ([12], Ch 7, using Parzen window) The numbers on the x-axis are the corresponding periodicities ----- marks the region of a periodicity of 5 days a No significant peaks in the spectrum b Positive correlation shows in the low frequency area c A peak at $1/\nu = 5$ indicates the presence of a periodicity of 5 days

time series (Figure 2c) but it is not as easily identified by means of an automated analysis as in the spectrum. Positive correlation ($0 < \alpha < 1$) gives a peak at low frequencies (Figure 3b) while negative correlation gives a peak in the spectrum at high frequencies. Like positive correlation, a trend shows a peak at low frequencies, but a trend also gives rise to non stationary behaviour. A test on stationarity [15] then has to distinguish trend from positive correlation. After the processes involved are qualitatively assessed, a suitable model is built. Consequently the model parameters can be estimated. These can be used to reconstruct the time series.

EXPERIMENTAL

AUTOCORR is written in SIMULA [16] and employs the expert system shell INFER. This simple shell makes depth-first backward reasoning possible. It was originally developed for LABGEN. Later, it was separated from LABGEN and made available for other SIMULA programs. Like AUTOCORR, the three other FEA programs and LABGEN are written in SIMULA.

Time series analysis requires much heuristic knowledge [12], and for this reason a knowledge based approach was chosen [17,18]. This approach implies the use of facts and rules. The facts in INFER are of the Attribute-Value type. The characterization of the time series consists of the determination of the values of the relevant facts. A rule is a structure to infer the value of a fact from other facts. In order to do so, the rule must be applicable, i.e. the conditions must be met. In the following section some examples are given. During the development of AUTOCORR new knowledge could easily be added to the structures present in INFER for rules and facts. Local changes did not interfere with the general course of the program.

On the other hand, the SIMULA environment makes algorithmic programming possible. This is useful for determination of the autocorrelation function, its Fourier transform and their characteristics. INFER

can use algorithmic procedures to obtain the value of a fact INFER generally uses the following strategy to find the value of a fact First it checks if the value of the fact is determined If this is not already done, INFER searches for an applicable rule If this path does not supply a value, INFER either consults the user or uses an algorithmic procedure to obtain a value This is predetermined by the nature of the fact, since algorithmic procedures can only be used in special facts called FACT_SPEC

An additional advantage of incorporating INFER is the presence of a user friendly man machine interface It provides uniform communication facilities for the LABGEN and FEA programs The few questions asked by AUTOCORR mainly concern periodicity If AUTOCORR detects a periodicity, it asks the user whether this is acceptable Other questions are avoided because these would require knowledge of time series analysis. This is contrary to the goal of the program.

Description of the program AUTOCORR

AUTOCORR consists of two knowledge bases (Figure 4) The program reads a time series and starts the analysis by consulting the main knowledge base. During the consultation the signal is tested on periodicity and INFER subsequently directs the analysis according to the outcome of this test If there is no periodicity the time series is treated as a whole The consultation of the main knowledge base is interrupted by the consultation of the second, sub knowledge base Now, the non-periodic processes (trend and autocorrelation) present in the time series are assessed With the result INFER resumes the consultation of the main knowledge base and proceeds with the estimation of the parameters

If there is a periodicity, the signal is split All Mondays are grouped in one time series, all Tuesdays in another, and so on Also the week totals are put in a new time series This week totals series is used to identify trend and autocorrelation effects These results are applied to

Figure 4 Representation of the major building blocks of AUTOCORR.

the week-day time series (mondays, tuesdays, etc.). This means that these series each have the same trend and autocorrelation characteristics. However, they may have different means and standard deviations.

The main knowledge base consists of 20 rules and 18 facts. The greater part is concerned with the elucidation of any periodicity and the rest with analysis control. Seven facts are of the type FACT_SPEC, four are used to calculate spectrum characteristics and three are used for control. For example, the fact PER15 is an instantiation of the FACT_SPEC CLASS PERIODLEVEL. After the algorithmic procedure has been exe-

cuted the value of PERI5 is equal to the contribution of the variance in that part of the spectrum which corresponds to a periodicity of five days (Figure 3)

The sub knowledge base contains 24 rules and 23 facts. Ten facts are of the type FACT_SPEC. This base is used to identify processes present in a time series which is under examination in the main knowledge base. First, significant correlation has to be established. If more than five percent outliers (based on eq. 4) are detected in the autocorrelation function, it is concluded that there is correlation present in the time series. But also if at low lag the autocorrelation function has significant high values there might be correlation.

An example of a rule is

```
IF OUTLIERS < 5 AND FRONTCORR > 1 AND FRONTCORR < 4
AND STAT = FALSE
THEN CORR = TRUE
```

This rule is for situations in which the presence of autocorrelation is not immediately clear. If either $\text{OUTLIERS} \geq 5\%$ (the fraction of outliers) or $\text{FRONTCORR} \geq 4$ (the number of positive outliers at the front of the autocorrelogram), it is immediately concluded from other rules that autocorrelation is present. If these conditions are not met, but there are some outliers at the front, there may be autocorrelation. This depends on other positive evidence. One of the rules stating this is the above-mentioned rule. In this case the additional positive evidence is obtained from a stationarity test of the time series, the result of which is available from the fact STAT. If this rule succeeds, usually a trend is present (this is to be positively concluded from other facts and rules).

During the consultation of the main knowledge base, data concerning the time series are recorded in a notebook structure. The notebook refers to a set of notes. Notes are instantiations of the CLASS NOTES, a framelike structure where a time series, its autocorrelation function and spectrum and associated parameters are stored. These attributes need only be calculated once. If there is no periodicity, the notebook refers to only one note, containing the original time series. Otherwise the note-

book refers to P notes each containing one of the week day time series. NOTES also has a reference to a template note. The template note serves to calculate the slope and autocorrelation coefficient of each series. NOTES has an internal procedure called `estimate_params` which estimates the parameters of its time series according to the model inferred.

Estimation takes place once a model is determined. The procedure `estimate_params` is triggered. First the slope of the trend is calculated. This is taken from the template time series, which is either the original series or the week series. Second the autocorrelation coefficient of the residual template series is determined by means of a linear least squares analysis. An exponential model is fitted to the autocorrelation function. The trend then is subtracted from the time series. The mean and overall variance are calculated from the residual series. Finally the standard deviation of the uncorrelated model is assessed. If a process is not present, the corresponding parameter is set to zero. All parameters are accessible in NOTES.

After both identification and estimation have taken place a report is written in the format requested by the input generator. Writing this file is the last action under INFER. Control is returned to the AUTOCORR main level. Here the session is terminated by writing the report to disk.

RESULTS

AUTOCORR has been subjected to an extensive test scheme. Time series of several types have been analysed by the program. A program generating time series according to the general model (eq. 1) has been written in SIMULA. A time series $\{x_t\}$ is generated from the following parameters: M_w , S_{Z_w} (with $w = 1$ or $w = 1, \dots, P$ in case of periodicity), Γ , α , N , SEED. N denotes the length of the time series and SEED the starting seed for the NORMAL random generator, a standard feature of SIMULA. If a process is not needed its parameter is set to zero.

In preliminary tests the absolute value of the mean did not influence the performance of AUTOCORR. This is in accordance with the theory concerning both stochastic and deterministic processes. An arbitrary variance level is chosen to elucidate the performance of AUTOCORR. The length of a laboratory data set usually ranges from 150 to 250 days, and a value $N = 200$ days is taken in most cases. Each parameter level is repeated ten times with different starting seeds.

Uncorrelated signal

First AUTOCORR is tested with time series containing only the uncorrelated process ($P = 1$, $\alpha = 0$, $T = 0$). 20 Different seeds were used in settings with $N > 50$ and $S_{Z,1} > 0$. In totally 100 different simulation runs AUTOCORR only identified the uncorrelated process and never assigned a deterministic or autocorrelation process.

Subsequently AUTOCORR has been tested for signals containing a combination of the uncorrelated signal with one of the three main processes: periodicity, trend or autocorrelation.

Periodicity

A cosine function is used to generate the mean M_w for the majority of periodicity tests. The amplitude A determines the contribution of periodicity to the whole signal (eq. 5).

$$M_t = A \cdot \cos \frac{2\pi(t-1)}{P} \quad (5)$$

The standard deviation $S_{Z,w}$ has the same value for each w . Detection of a periodicity in the simulated signal is hardly dependent on the length of the series, if the period is at least approximately ten times shorter than

Figure 5. Fraction of identified processes as a function of the amplitude of the cosine function (eq. 5). Only an uncorrelated and a periodic process are generated. Each fraction is determined from 10 different runs ($P = 5, 7, T = 0, \alpha = 0, N = 200, M, (eq. 5), S_{Z,w} = 20$)

the series length. Both a five and a seven day periodicity are equally well detected. Detection strongly depends on the magnitude of the standard deviation. Given the cosine function the periodicity is identified if A is globally larger than $S_{Z,w}/2$.

For $N = 200$ and $S_{Z,t} = 20$ the performance of AUTOCORR is examined in this way: for $A = 0, 5, \dots, 20$ (samples) ten time series were generated and examined by AUTOCORR. The fraction of identified periodicities was determined, as is shown in Figure 5. Only periodic processes were identified. For both $P = 5$ and $P = 7$ the periodic process is identified in 7 out of 10 cases at $A = 10$. The identification limit, defined as the value beyond which a process will be identified with more than 95% certainty, lies between $A = 10$ and $A = 15$ (samples).

Since S_M , the standard deviation of the mean values M_w for the week-days, is proportional to the amplitude of the cosine, it can also serve to describe the identification limit. This is also done for other cycl-

Table 1

The identification limit of the cyclic functions represented in Figure 6. The standard deviation of the mean values of the week-day series, S_M , expresses the identification limit. Each identification limit is determined in five simulation runs ($P = 5$, $T = 0$, $\alpha = 0$, $N = 200$, $S_{7w} = 20$).

Figure	function	identification limit S_M
6 a	cosine	5.8
6 b	single step	6.3
6 c	spike	6.8
6 d	triangle	6.8
6 e	double spike	not detectable

ic functions such as a single step or spike (Figure 6). Each has a different identification limit as can be seen in Table 1. AUTOCORR is capable of detecting most shapes. Some will not be detected, such as a double spike (Figure 6e). In the latter case the spectrum does not show any five day correlation.

Trend

The smallest value of slope T at which a trend is identified is dependent on both the standard deviation and the length of the simulated signal. If the standard deviation of the uncorrelated random process is large, it is difficult to detect a small trend. However, the longer the time series, the easier a small trend can be identified.

Figure 6. Examples of cyclic functions used in simulating periodic processes. Periodicity: 5 days. a. cosine function; b: single step function; c: spike function; d: triangle function; e: double spike function

Figure 7 Fraction of identified processes as a function of the slope T . Only an uncorrelated process and a trend are generated. Each fraction is determined from 10 different runs ($P = 1$, $\alpha = 0$, $N = 200$, $M_1 = 40$, $S_{Z,1} = 20$).

For $N = 200$ and $S_{Z,1} = 20$ the performance of AUTOCORR is examined as above. The range of the slope was $T = 0$ to 0.20 (samples/day). The latter causes a total shift in the value of the mean of 40 (samples). In all cases an uncorrelated random process either with or without trend is found, and no other processes were identified in the simulated signals (Figure 7). The identification limit lies at $T = 0.15$ (samples/day).

Autocorrelation

The autocorrelation process is detected by AUTOCORR regardless of the value of $S_{Z,w}$. The length of the time series is important. Although the simulated autocorrelation is detected in both long and short series, in short series more frequently an additional trend is identi-

Figure 8 Fraction of identified processes as a function of the autocorrelation coefficient α . Only an autoregressive process is generated. Each fraction is determined from 10 different runs ($P = 1$, $T = 0$, $N = 200$, $M_1 = 40$, $S_{Z,1} = 20$)

fied. From one specific realisation of a stochastic process it is difficult to discriminate between autocorrelation on one hand and autocorrelation and trend on the other hand. Only the analysis of several time series originating from the same stochastic model makes it possible to determine whether a trend is or is not present. AUTOCORR has only one time series for analysis.

For $N = 200$ and $S_{Z,1} = 20$ the performance of AUTOCORR is examined for α at levels 0.2, 0.4, 0.6, 0.8 and 0.9, corresponding to a time constant of resp. 0.5, 1, 2, 4, 9 and 19 days (Figure 8). At $\alpha = 0.2$ and $\alpha = 0.4$ only autocorrelation is identified. The higher the autocorrelation value the more often also a trend is attributed to a time series, although the trend actually is not present.

More complicated signals are also handled by AUTOCORR. Two processes are combined with the uncorrelated process in one time series. The parameters have values close to the identification limit of the pro-

Figure 9 Fraction of identified processes in simulated series containing an uncorrelated process, a trend and a cosine, as a function of the slope T and the amplitude of the cosine A . Each fraction is determined from 10 different runs ($P = 5, 7$, $\alpha = 0$, $N = 200$, M_w (eq. 5), $S_{f_w} = 20$).

Periodicity and trend

Figure 9 shows the results for signals with a periodicity and a trend. The values on the axes belong to signals with only one process; these are discussed above. The periodicity levels are $A = 10, 15$ (samples) and the trend levels $T = 0.10, 0.15, 0.20$ (samples/day). The fraction trend identified is larger if periodicity is present. This is due to the fact that, once periodicity is found, the week-day results are put together, thus enhancing any increase or decrease in mean. The presence of a trend suppresses the identification of a periodicity slightly. There is hardly any difference in detection of a five or a seven day period. When the parameters are at high level both periodicity and trend are identified.

Figure 10 Fraction of identified processes in simulated series containing an autoregressive process and a cosine, as a function of the amplitude of the cosine A and the autocorrelation coefficient α . Each fraction is determined from 10 different runs. ($P = 5$, $T = 0$, $N = 200$, M_w (eq 5), $S_{Z,w} = 20$).

Periodicity and autocorrelation

Figure 10 shows the results when periodicity and autocorrelation processes are present in the time series. The levels are shown for $A = 5$ samples and $\alpha = 0.4, 0.7$ and 0.9 . Now the periodicity is found at much lower level: periodicity is identified in all 10 cases at level $A = 5$ (and higher). As is shown in Figure 8, at this level AUTOCORR did not identify periodicity at all for a signal with only a periodic process. This difference is mainly caused by interaction in the spectrum. The presence of a first order autoregressive process also contributes to the variance in the periodicity region. In spite of the interaction, AUTOCORR detects the true period.

Figure 11 Fraction of identified processes in simulated series containing an autoregressive process and a trend as a function of the slope T and the autocorrelation coefficient α . Each fraction is determined from 10 runs with different seeds ($P = 1, N = 200, M_1 = 40, S_{Z,1} = 20$)

AUTOCORR does not detect autocorrelation in all cases at $\alpha = 0.4$, contrary to the case when only autocorrelation is present (cf. Figure 8). At higher levels it always identifies autocorrelation. AUTOCORR then also attributes occasionally a trend. This resembles the case when only autocorrelation is simulated.

Trend and autocorrelation

Figure 11 shows the results of combining trend and autocorrelation processes. Trend is at levels $T = 0.5, 0.10$ and 0.15 (samples/day) and autocorrelation at levels $\alpha = 0.4, 0.7, 0.9$. The autocorrelation process is always identified here. The trend is identified more often at a higher slope level but even at these high levels there is a fraction where only autocorrelation is identified and not the trend. As seen above, in a pure autocorrelation process a trend can also be identified by AUTOCORR. Probably the trend in the autocorrelation process counteracts the trend process in the time series simulated. At high T level the trend will become more dominant and both trend and autocorrelation will be identified.

Estimation

If AUTOCORR correctly identifies the processes present in the simulated time series, it estimates the relevant parameters with a relative precision of approximately 10%. In two situations, however, AUTOCORR sometimes does not assign the right processes to the time series. These are: either a trend is identified in a trendless series or a trend is not identified where one is present. In both cases the estimated slope value does not resemble the real value. The autocorrelation coefficient, on the other hand, still resembles the real value rather well.

DISCUSSION

Two criteria are useful to judge the performance of AUTOCORR. The false positive identification, indicating a process that is not present, and the false negative identification, not detecting a process that is present. These criteria are qualitatively applied to the three main processes.

False positive identifications are not probable. Only in the case of an autocorrelation process with a large α it is sometimes seen that a trend is erroneously identified. Because a manual time series analysis would yield the same result, this is not a real failure of AUTOCORR but an intrinsic problem of time series analysis. Nevertheless, the value of the estimated slope can sometimes be rather large at high autocorrelation coefficient values. Neither periodicity nor autocorrelation processes were ever falsely identified in the 200 resp. 220 simulation runs in which they were not present. AUTOCORR performs very well in this respect.

False negative identification occurs when a process is present under its identification limit. This limit is fairly low, judged by the comparison of the standard deviation of the whole time series to that of the uncorrelated process. If the former is only 10% larger than the latter, already the identification limit is met. This goes for all three processes.

By lowering the identification limits, the amount of false negative identifications would decrease but simultaneously the amount of false positive identifications would increase. This is not acceptable. Applying the results to reconstruct a time series in order to simulate the operation of a laboratory, the effects of a false positive identification are more harmful than a false negative identification. With the present identification limits AUTOCORR only incorporates the dominant processes and correctly models the laboratory signal. In the worst case no improvement is made by using AUTOCORR.

Extension of AUTOCORR is possible but not necessary. The periodicity knowledge base can easily be extended so that other periodicities can be found in the range from $P = 3$ to $P = 14$. This has been successfully tested but was not incorporated in the final program, because the

most probable cycle in a laboratory is the working week. A monthly period is difficult to identify because it interferes strongly with trend and autocorrelation processes and because the laboratory signal usually is too short to comprise enough periods.

CONCLUSION

AUTOCORR successfully analyses a time series and gives a good description of it by estimating its parameters. Although the analysis is by no means as elaborate as the Box and Jenkins approach [14], it is sufficient at least for application in a simulation context. This means that the results of the analysis are used to reconstruct the input signal in order to simulate the operation of an analytical laboratory. Only first order approximations are used. General applicability to all kinds of signals is not pursued. The program is specific for laboratory input signals which implies the analysis contains assumptions that are only valid in this domain. However, this approach allows quite complicated time series comprising different processes to be adequately described. The essential characteristics are captured: autocorrelation, trend and periodicity. In most cases these processes are correctly identified by AUTOCORR. After identification it builds a parsimonious model, where parameters of absent processes are set to zero.

In time series analysis extensive calculations must be combined with decision making based on experience. An important reason for the successful implementation of AUTOCORR is the fact that INFER offers the possibility to combine "conventional" programming for the calculations with the reasoning structures for decision making. In analytical chemistry there are many other domains where calculations must be combined with decision making, e.g. calibration and statistical testing in non-routine situations. It is important to realize that many commercially available empty shells for knowledge representation are not suitable for these domains because they have only limited possibilities with respect to mathematical operations [19]. Only those shells can be successful in

these domains that either allow a link to a calculation medium or are able - like INFER - to incorporate the "conventionally" programmed procedures

REFERENCES

- 1 B G M Vandeginste, *Anal Chim Acta*, 122 (1980) 435
- 2 P Winkel, *Clin Chem*, 30 (1984) 1758
- 3 T A H M Janse, G Kateman, *Anal Chim Acta*, 159 (1984) 181
- 4 B van de Wijdeven, J Lakeman, J Klaessens, B Vandeginste, G Kateman, *Anal Chim Acta* 184 (1986) 151
- 5 J Klaessens, T Saris, B Vandeginste, G Kateman, *J Chemometrics*, 2 (1988) 49
- 6 J Klaessens, L van Beysterveldt, T Saris, B Vandeginste, G Kateman, LABGEN, expert system for knowledge based modelling of analytical laboratories Part I Laboratory organization, *Anal Chim Acta*, submitted for publication
- 7 J Klaessens, J Sanders, B Vandeginste, G Kateman, LABGEN, expert system for knowledge based modelling of analytical laboratories Part II Application to a laboratory for quality control *Anal Chim Acta*, submitted for publication
- 8 J Klaessens, J van Schalkwijk, P Cox, R Bezemer, B Vandeginste, G Kateman *J Chemometrics* 3 (1988) 81
- 9 D L Massart, B G M Vandeginste, S N Deming, Y Michotte, L Kaufman, *Chemometrics a textbook* Elsevier, Amsterdam, 1988
- 10 P J W M Muskens, W G J Hensgens, *Water Research*, 11 (1977) 509
- 11 G M Jenkins, G van Osch, Relationship between ammonia concentration, flow rate and temperature in the river Rhine", in G M Jenkins, "Case studies in time series analysis" Elsevier Amsterdam, 1982
- 12 C Chatfield, *The analysis of time series*, 2nd ed, Chapman & Hall, London, 1980
- 13 H C Smit, Autocorrelation and time series analysis, in B R Kowalski (ed.), *Chemometrics, mathematics and statistics in chemistry*, Reidel Publ Co, Dordrecht Holland, 1984 p 157 176
- 14 G E P Box, G M Jenkins, *Time series analysis Forecasting and control*, Holden-Day, San Francisco, 1976
- 15 L Sachs, *Angewandte Statistik* 6. Aufl Springer, Berlin, 1986, p 209
- 16 G M Birtwistle, O J Dahl, B Myhrhaug, K Nygaard, Simula Begim, Petrocelli/Charter, New York, 1973
- 17 J Klaessens, G Kateman, *Fres Z Anal Chem*, 326 (1987) 203
- 18 R E Dessy (ed.), *Anal Chem*, 56 (1984) 1200A, 1312A
- 19 N G van der Gaast, L H de Jonge, *Trends Anal Chem*, 7 (1988) 283

PROGRAM FOR PROCESSING OF HISTORICAL LABORATORY DATA BASED ON FUZZY SET THEORY AS A TOOL FOR DECISION SUPPORT

J. KLAESSENS, J. VAN SCHALKWIJK, P. COX, R. BIZEMER, B. VANDEGINSIT AND
G. KATIMAN

*Laboratorium voor Analytische Chemie, University of Nijmegen, Toernooiveld VI 6525 ED Nijmegen,
The Netherlands*

SUMMARY

A computer program (FLA) is presented for processing historical laboratory data. It performs on a list of sample entries stored in a laboratory information management system. Using an algorithm which is based on fuzzy set theory, FLA classifies the entries into a limited number of clusters called simple types. The classification is fully user-defined. The program transforms the historical data into a representation which is more suitable for studying the performance of the laboratory or which can be used as preparation for a simulation project.

KEY WORDS: Fuzzy set theory, Classification, Data reduction, Historical laboratory data

INTRODUCTION

Because analytical laboratories typically process large numbers of samples every day, the management of these samples and the subsequent test results is of critical importance. The need for suitable software tools to facilitate this management is evident and has been recognized. For several years laboratory information management systems (LIMS)¹ have been available commercially from different suppliers and implemented in an increasing number of laboratories. Many laboratories have decided to develop their own LIMS.

Essentially, a LIMS is a data base to store data about samples and a number of functions to guide the storing and retrieving of these data. Some major advantages of the application of a LIMS are: the laboratory manager can easily track the status of any sample at any time; specific reports can be generated automatically; historical test results are easily accessible for e.g. trend monitoring. The data stored in a LIMS reflect the character of the laboratory in the way it responds to the needs of a larger organization. As a consequence, it would be very interesting to be able to employ these data to solve management problems. However, it must be realized that a LIMS contains raw data, not information which is suitable to be employed immediately for this purpose.² The raw data must be refined.

Current LIMS have functions to retrieve all sample entries with some specific property from the data file. For management purposes, however, it is often more important to be able to study groups of sample entries which are related for a particular property instead of identical

0886-9383/88/050081-15\$07.50

© 1988 by John Wiley & Sons, Ltd.

Received 30 March 1988

Revised 28 June 1988

For instance, from the management point of view, the question 'How well does the laboratory perform with respect to heavy metal analyses?' will usually be more important than 'How well does it perform with respect to Cd analyses?' A first step to increase the values of a LIMS as a tool for laboratory management (and thus becoming more than a tool for sample information management) is to incorporate functions or modules for this clustering analysis. These will support the manager while investigating the performance of the laboratory.

A second step of course would be modules that support the manager while making decisions to improve this performance. Because digital simulation is a powerful method for decision support^{3,5}, a promising approach is to incorporate simulation modules into the LIMS environment. These will enable the manager to construct a simulation model of the laboratory and carry out simulation experiments in a reasonably short time without expertise in simulation being necessary.⁶ Data necessary for a simulation model can be collected partly from a LIMS. In most cases, however, it will be necessary to perform a refinement of the raw data first.

The present paper describes a computer program which classifies the sample entries and their properties into a limited number of clusters called sample types. Although the program is predominantly described in the context of a subsequent simulation project, it is applicable generally for analysis of historical laboratory data.

THEORY

Sample types and their properties

A sample type may be defined as a class of sample entries having similarities in the way they are processed in the laboratory. These are similarities not necessarily in a chemical sense, but in a sense of interactions caused in the laboratory. Examples of similarities which are relevant for laboratory management are: processed by the same technician(s) or on the same instrument(s); similar pattern of sample processing times (sample preparation and measurement).

The historical data file of the LIMS contains a number of data fields for each sample entry. Some data fields are not relevant for the classification into sample types, e.g. the chemical result of the test or the request code of the sample. In many instances the sample name or sample code suffices for a first classification. The values of the other data fields are stored in data structures (e.g. histograms) of the sample type. The result of the classification is a set of sample types, each having a number of properties (i.e. the stored value of the data fields). Therefore the procedure is called type construction. Two kinds of properties can be distinguished: numerical, e.g. batch size, processing times; alphanumerical, e.g. sample name, technician who carried out the analysis. An alphanumerical property is a histogram of values (e.g. technician names) and corresponding frequencies of occurrence.

Mostly, this set of sample types is not yet suitable for use in a simulation project because it does not meet the following constraints: the frequency of occurrence of the sample types must be adequate; low frequencies tend to render the simulation model rather unstable; the number of sample types may not be too large (about 20)—a large number is difficult to handle in a simulation model and causes difficulties in interpretation. Similar considerations apply when the classification is used as a tool on its own and is not followed by a simulation project. In order to fit the set of sample types to these constraints, a type reduction must be performed, i.e. groups of two or more related sample types are combined into new ones, called meta types.

The problem is how to reduce the set of sample types, which is a pattern recognition problem. Recently, an approach has been reported which is based on factor analysis.⁷ A set of abstract sample entry vectors is calculated. In the simulation model the sample entries for each day are determined as a linear combination of these vectors. This approach is considered to be too abstract to form the basis of a generally applicable program which—unavoidably—requires interference of the user (i.e. the laboratory manager). An intuitively more appealing approach is the application of fuzzy set theory.

Some fundamentals of fuzzy set theory

Fuzzy set theory was introduced by Zadeh⁸ and may be considered as a generalization of conventional set theory. Important applications of fuzzy set theory in analytical chemistry have been reported, e.g. for multicomponent spectral analysis,⁹ and have been reviewed by Bandemer and Otto.¹⁰

In conventional set theory each element of a space X either belongs to a specific set A ($A \subset X$) or not. Fuzzy set theory introduces a continuum of grades of membership to a specific fuzzy set A . If $x_1, \dots, x_n$ are elements of X , a fuzzy set A of X is a set of pairs

$$A = \{x_i, f_A(x_i)\} \quad f_A(x_i) \in [0, 1] \quad (1a)$$

or, in short, the fuzzy set A is characterized by the membership function

$$f_A: X \rightarrow [0, 1] \quad (1b)$$

Here $f_A(x_i)$ represents the grade of membership of element x_i to the fuzzy set A . The grade of membership varies between $f_A = 1$ and $f_A = 0$; these extreme values correspond to $x_i \in A$ and $x_i \notin A$ respectively in conventional set theory.

Although the membership function may be defined by a probability function, it must be realized that its basic nature is not statistical. The probability function is objective because it can be determined from a series of experiments. In essence, the membership function is subjective; it is used to incorporate subjective experience. It is also dependent on the context in which it is used. Fuzzy sets are very suitable for describing imprecisely defined notions such as 'large' and 'young'. The membership function of 'large' in the context of organic molecules will differ greatly from that in the context of DNA molecules.

Many operations are defined to manipulate fuzzy sets.¹¹ The operations that are relevant to the type reduction problem will be considered in the next section.

Application of fuzzy set theory to type reduction

The result of the type construction is a set of sample types, each having a number of numerical and/or alphanumerical properties. In order to select the groups of related sample types, the values of the type-determining properties are interpreted as fuzzy sets. It is not relevant whether values of a property are equal, but whether they support the possibility of merging the two sample types. This depends on the kind of property, the laboratory context and even the goal of the analysis. In order to guarantee the required specificity, the degree of fuzziness of each property is defined by the user. Two cases are distinguished:

Figure 1 The membership functions of a property for two sample types $T1$ and $T2$ (having values $v1$ and $v2$). The intersection of the two fuzzy sets is denoted by $A12$.

Numerical property

The value of a numerical property is fuzzed by introducing a membership function around this value. Figure 1 shows the membership function of the fuzzy sets $A1$ and $A2$ of two sample types $T1$ and $T2$ with respect to a given property (with the values $v1$ and $v2$). The possibility of merging $T1$ and $T2$ is supported by the intersection $(A1 \cap A2)$ of the two fuzzy sets. The membership function of the intersection is defined by

$$f_{A1 \cap A2}(x) = \min\{f_{A1}(x), f_{A2}(x)\} \quad (2)$$

The larger the intersection, the more the possibility of merging is supported. In order to consider more than two sample types with respect to one property, intersections must be compared. Consider, for instance, sample types $T1$, $T2$ and $T3$ (Figure 2). Merging of $T1$ and $T3$ is not supported because $A1 \cap A3 = \emptyset$. Merging of $T2$ and $T3$ is supported more strongly than that of $T1$ and $T2$. Although both intersections contain different elements, a measure to compare them is provided by the power of a fuzzy set. This operator is defined by

$$p(f_A) = \sum_{x \in X} f_A(x) \quad \text{for a finite space } X \quad (3a)$$

$$p(f_A) = \int_X f_A(x) dx \quad \text{for an infinite space } X \quad (3b)$$

The power operator can be considered as the fuzzy counterpart of counting the number of elements of a conventional set. In order to allow comparison between properties, the relative power of the intersection is applied. Henceforth this entity will be referred to as measure of support.

$$MS = p(f_{A1 \cap A2})/p(f_{A1}) \quad (4)$$

Figure 2 The membership functions of a property for three sample types $T1$, $T2$ and $T3$ with values $v1$, $v2$ and $v3$. The intersections are denoted by $A12$ and $A23$.

The fuzzy sets have a triangular shape (Figures 1 and 2) because this allows easy calculation of MS. It can easily be demonstrated that in that case

$$MS = h^2 \quad (5)$$

where h denotes the height of the intersection triangle. Because in the type reduction only a comparison of MS values is pursued, it is unlikely that the exact shape of the fuzzy sets interferes with the results.

The spread of the fuzzy set is very important. The larger this spread, the larger will be MS given two values of a property. This value incorporates the importance of a property with respect to merging two sample types. Therefore the user should supply this value for each type-defining property. In order to improve the ease of application,² the user may choose from five levels of resemblance requested for the property (Table 1). The chosen level is translated into fuzzy sets by using the keys of Table 1.

Table 1. Levels of resemblance requested for a numerical property and the spread of the corresponding fuzzy set as a percentage of the allowed region of the property.

Level of resemblance requested	Spread of fuzzy set (%)
Complete	1
Considerable	11
Intermediate	22
Moderate	33
Weak	50

It must be realised that this approach is of a non-statistical nature and fully user-determined. The user can perform an analysis which is statistically meaningless. By defining the spread of the fuzzy sets smaller than the statistical dispersion of the property, low values of MS may be obtained whereas the values do not significantly differ. The condition that the fuzzy set should be broader than the statistical distribution cannot be required. It is ambiguous because mean values are compared based on different numbers of observations and as a result—having different statistical distributions. It is much more appealing to apply the simple fuzzy approach and to allow user control for meaningfulness. This is not a very critical demand because choosing the fuzzy set too narrow results in too many sample types being constructed, which is likely to be corrected by the user.

Alphanumerical property

If a property is of an alphanumerical nature, a different procedure must be followed. This property consists of a histogram with a number of values and frequencies of occurrence. Now the normalised histogram (relative frequencies) itself is interpreted as a fuzzy set. The MS value for this property is calculated with equations (2), (3a) and (4) taking into account that the values of the alphanumerical histograms cannot be ordered.

Figure 3. Various types of manipulation for calculation of MS for the same pair of values (i.e. histograms) of an alphanumerical property. One histogram (full lines) contains objects A, B, C, D and E, the other (dashed lines) contains B, D, E and F. MS is calculated as the overlap of the histograms (shaded). (a) No manipulation. (b) actual frequencies are less important for comparison manipulation $\text{dil} (n = 2)$. (c) comparison is predominantly based on objects with high frequencies manipulation $\text{con} (m = 2)$. The calculated MS values are listed in Table 2.

The different character of the property and its membership function requires a different kind of user interference. There are more ways to manipulate the sensitivity of MS with regard to the actual value of the property (i.e. histogram). Two are very useful in the context of sample types.

- (1) The mere presence of the values in the histogram is more important than their frequencies. This may be the case when the instruments for the sample types are being compared. Independently of its frequency, the presence of the instrument in the histogram implies that the sample can be analysed on it. The dilation operator can be used to perform this manipulation.

$$\text{dil}(f_A(x_i)) = f_A(x_i)^{1/n} \quad (n > 1) \quad (6)$$

The operator is applied on the fuzzy set and followed by a renormalization. Depending on the value of n , MS becomes less dependent on the frequencies, as shown in Figure 3.

- (2) Sometimes it is more appropriate to consider only the values of the histogram with the highest frequencies. If the histograms of analysts for the sample types are compared, it may be useful to pay little or no attention to analysts who only incidentally analyse a sample of this type. In this case the concentration operation can be used in a similar manner as the dilation operation.

$$\text{con}(f_A(x_i)) = f_A(x_i)^m \quad (m > 1) \quad (7)$$

An example of the result of this operation is shown in Figure 3.

With respect to the calculation of MS for an alphanumerical property, the user is allowed to choose from five levels of resemblance requested (Table 2). This table also contains MS calculated for the data of Figure 3.

If two or more properties are type-determining, an overall measure of support MS^o for each interaction of two sample types is calculated. It is determined as the most restrictive value of the separate MSs.

$$\text{MS}^o = \min_i \{\text{MS}_i\} \quad (8)$$

where i lists all type-determining properties.

Table 2 Types of resemblance requested for an alphanumerical property the corresponding manipulation of the fuzzy set and MS calculated for the property of Figure 3

Resemblance requested	Type of manipulation	MS calculated
Almost only values	dil ($n = 3$)	0.54
Values stressed	dil ($n = 2$)	0.51
Intermediate	—	0.39
High frequencies stressed	con ($m = 2$)	0.22
Almost only high frequencies	con ($m = 3$)	0.12

For each combination of two sample types MS' is calculated. The combinations with the largest MS are merged into meta types until a suitable number of sample types is obtained or MS becomes too small. The user is able to interfere with this automatic process e.g. to undo or add certain combinations.

DESCRIPTION OF THE PROGRAM FEA

The first implementation of the FEA (front end analysis) program was written in the language FRANZ LISP¹¹ and run on a VAX 11/750 with VMS operating system. This language was chosen because of its very flexible data structures such as lists and property lists. These are very suitable for a program to be applied to many different data files of different laboratories. Although the calculations were performed by external FORTRAN subroutines linked to the LISP structures, the program appeared to be too slow to handle data files with several thousand sample entries. To some extent this problem was caused by the fact that only parts of the program could be compiled and for other parts the interpreter had to be used.

In order to speed up the program a second implementation was written in SIMULA and run on a NAS 9600 mainframe under the VM CMS operating system. Using SIMULA's class

Figure 4 The three parts constituting the FEA program

concept and dynamic queues,^{14, 15} it was possible to model this second implementation in a similar manner to the LISP version. A second advantage of SIMUL A is that this language is also used by the program LABGEN and the simulation models generated by LABGEN.⁶ As shown in Figure 4, FEA consists of three parts.

Preprocessing

The first program of FEA performs a number of preparation activities, most of which are optional. Some of these activities result in a rearranged data file.

- (1) If the data file consists of data fields on fixed positions, the data fields are separated by blanks. This latter format is necessary for the type construction program. During this rearrangement process data on certain positions may be omitted.
- (2) If samples arrive at the laboratory in batches, an analysis for batch entrance is appropriate. This means that a number of successive sample entries with equal properties are considered to belong to one batch entry. In the rearranged data file they are recorded as a single entry with the batch size as an additional data field.
- (3) If a data field consists of a variable number of items, each having significance of its own, the frequencies of occurrence of the items can be analysed and, if necessary, the items can be manipulated. The information of the data field is recorded in the rearranged data file with a special format that is recognized by the type construction program. An example of such a data field is the list of tests which are performed on the sample.

It is not exceptional that a laboratory receives samples from different departments or clients and that these samples, although physically equal, are processed with different priorities. In these cases it is useful to construct a simulation model with more than one sample generator. The necessary data are collected by the preprocessing program. If this analysis is requested, the user should specify the data field which contains the information about the sample origin. For each distinct origin the characteristics of the input distribution are recorded, optionally for each weekday.

Type construction

The type construction program classifies the sample entries of the (rearranged) data file into a number of sample types. To initialize the program, the user should provide information about

- (1) which data fields—if any—are to be ignored, their values are not stored as properties of the sample types.
- (2) which data fields are stored in alphanumerical histograms.
- (3) which data fields are of a numerical nature, either their values may be stored in numerical histograms or only their statistical characteristics may be calculated.
- (4) which data fields contain the time characteristics of a sample entry, e.g. date or time of entrance, if present, they are used to calculate the inter-arrival times and the delay times of the samples.
- (5) which data fields contain several independent items, with respect to type construction the total is considered as a single value, but the items are stored independently in an alphanumerical histogram.
- (6) which data field(s) is (are) used for type construction, each different combination of values results in a new sample type.

After the initialization, the program processes all sample entries. This main loop is shown in Figure 5. Each sample type is represented as an object of a SIMULIA class which contains slots to identify the sample type, to store the number of entries and to store the date time characteristics of the last entry. It also contains a dynamic queue with the properties of the sample type, i.e. a queue of alphanumerical or numerical histograms. The value of each relevant data field is stored in the corresponding histogram.

The result of the construction phase is a list of sample types and their properties. The program offers several facilities to inspect the sample types in order to decide whether the type construction is useful and whether a subsequent type reduction is necessary. At this point the user is also allowed to discard less frequently occurring sample types.

Figure 5 Flow diagram of the main loop of the type construction program of FEA

Type reduction

If the number of sample types is considered to be too large, a type reduction may be carried out. This program processes a list of sample types and can only be applied after the type construction has been completed. The information requested from the user is which properties are to be considered for the reduction process and how must MS be calculated for each of these properties (see Tables 1 and 2).² Alphanumerical properties can also be

Figure 6. Main activities and data representation of the type reduction program.¹ By combining the properties of the sample types of the meta set.

compared in a crisp way: $MS = 1$ only if both histograms are equal; otherwise $MS = 0$. The main activities and data representation of the program are shown in Figure 6.

For each pair of sample types MS^0 is calculated. If it exceeds a limiting value (which is to be entered by the user), an object of a SIMULA class is created containing this value and the two sample types. This object is stored in an ordered queue. Using the largest MS^0 first, meta types are proposed. At first, each proposed meta type is merely a queue of the constituent sample types. After the user's approval, final meta types are constructed, i.e. all sets of properties are combined into a single set of properties.

Type reduction is an interactive process: the user must find suitable program settings in order to create a satisfactorily small number of sample types. Therefore the user is enabled to inspect all meta-types and—if necessary—to undo a former step. It is also possible to create meta types by hand or to delete types (sample types or meta types) altogether. The final types and their properties are printed out.

APPLICATION

The FEA program has been tested with some data files from former simulation projects.^{4,6} These tests were performed to debug and validate the program. They resulted in reasonable

sets of meta types. In this section an application will be described in more detail. The data analysis with FEA is carried out as a part of a simulation project using the program LABGEN for automatic model construction. The simulation part will be reported later.

The laboratory considered is the heavy metals section of a laboratory for quality control of drinking water. Increasingly, samples from external clients are analysed. In the section samples are analysed on two AAS instruments (flame and furnace). Depending on the workload, the section is staffed with two to four technicians. The data file contained all sample entries (about 2500) from 19 March until 30 June 1987 in chronological order and were taken from an in-house LIMS. No additional programming was needed to collect the data in a file suitable for use by FEA.

Preprocessing of the data file

The datafile contains nine data fields for each sample. Two data fields are ignored during preprocessing: sample number and sampling point. Figure 7 shows the remaining seven data fields of the first part of the data file. These data fields contain information about

- (1) sample origin (client) (column 1 in Figure 7)
- (2) dates of entrance in laboratory and reporting (columns 2 and 4)
- (3) type of sample pretreatment and tests (columns 6 and 7), the tests are represented by codes of which the first two letters denote the element to be determined, e.g. CUG means Cu^{2+} determination
- (4) other sections of the laboratory visited (column 5)
- (5) type of sample (water, soil) (column 3)

1	2	3	4	5	6	7	i	column
2	190387	3	300387	-	13	CUG_CDG_AS		
2	190387	3	300387	-	13	CUG_CDG_AS		
2	190387	3	300387	-	13	CUG_CDG_AS		
2	190387	3	300387	-	13	CUG_CDG_AS		
2	190387	3	300387	-	13	CUG_CDG_AS		
2	190387	3	300387	-	13	CUG_CDG_AS		
2	190387	3	250387	A_O	13	CUG		
2	190387	3	300387	O_A	1314	CUG_CDG_FBG_AS_HG_ZNG		
2	190387	3	300387	A_O	1314	CUG_CDG_FBG_AS_HG_ZNG		
2	190387	3	300387	O_A	1314	CUG_CDG_FBG_AS_HG_ZNG		
2	190387	3	300387	A_O	1314	CUG_CDG_FBG_AS_HG_ZNG		
2	190387	3	300387	O_A	1314	CUG_CDG_FBG_AS_HG_ZNG		
2	190387	3	010487	A_O	1314	CUG_CDG_FBG_CRG_AS_HG_ZNG		
2	190387	3	010487	O_A	1314	CUG_CDG_FBG_CRG_AS_HG_ZNG		
2	190387	3	010487	A_O	1314	CUG_CDG_FBG_CRG_AS_HG_ZNG		
2	190387	2	010487	O_F	0	CU_CD_FB_CR_AS_HG_ZNG		
2	190387	2	010487	O_F	0	CU_CD_FB_CR_AS_HG_ZNG		
2	190387	3	020487	A_O	1314	CUG_CDG_FBG_CRG_AS_HG_ZNG		
2	190387	3	020487	O_A	1314	CUG_CDG_FBG_CRG_AS_HG_ZNG		
2	190387	3	020487	A_O	1314	CUG_CDG_FBG_CRG_AS_HG_ZNG		
2	190387	3	020487	O_A	1314	CUG_CDG_FBG_CRG_AS_HG_ZNG		
2	190387	2	020487	F_O	0	CU_CD_FB_CR_AS_HG_ZNG		
2	190387	2	020487	F_O	0	CU_CD_FB_CR_AS_HG_ZNG		
2	190387	2	020487	O_F	0	CU_CD_FB_CR_AS_HG_ZNG		
2	190387	3	130487	O_F_A	1314	CUG_CDG_FBG_CRG_AS_HG_ZNG		

Figure 7 The first part of the historical laboratory data file after removal of two data fields. The information stored in the data fields is described in the text.

Table 3 Mean values of batch entries per day specified according to weekday and sample origin. The values in parentheses represent standard deviation

	Total		0		1		2*	
Week	10-69	(6.1)	3.99	(3.3)	1.31	(1.6)	5.39	(4.3)
Monday	8.80	(6.2)	2.00	(1.7)	1.20	(1.9)	5.60	(4.9)
Tuesday	12.47	(3.1)	7.13	(2.4)	0.87	(1.1)	4.47	(2.2)
Wednesday	16.14	(4.7)	7.43	(1.5)	2.14	(1.5)	6.57	(5.4)
Thursday	10.80	(6.5)	3.07	(2.3)	1.47	(1.9)	6.27	(4.9)
Friday	5.60	(4.3)	0.53	(0.8)	0.93	(1.3)	4.13	(3.8)

*A combination of seven sample origins

The samples are mainly characterized by the chemical tests requested. Therefore it is useful to precede the type construction by an analysis of the tests and the frequencies of occurrence. Apart from sample pretreatment, 69 different tests were present in the data file, some of which occurred only once. Based on chemical information, several tests could be combined because there were only minor differences. The remaining infrequent tests were combined into two rest-groups, depending on the instrument. The result of this manipulation was 21 tests, which is acceptable for a simulation model.

The next step of the preprocessing is the analysis for batch arrival. Successive samples in the data file, which enter the same day and have the same sample origin, pretreatment and tests are considered to belong to the same batch. By this operation the data file is transformed from about 2500 individual sample entries into approximately 800 batch entries with a mean batch size of three.

Analysis of the input stream of samples resulted in nine distinct sample origins (which are in fact groups of similar internal or external clients), six of which generate relatively few samples. As a first approach these six origins were combined. However, at a later stage it appeared that for modelling purposes the six infrequent origins could best be combined with one of the frequent origins. The result is three distinct sample origins. For each origin a histogram of batch entries per day was calculated for each weekday. The mean values and the standard deviations are listed in Table 3. The laboratory input is dependent on the weekday.

Type construction and reduction of the data file

A type construction was performed based on the alphanumerical properties sample pretreatment and tests. The data file contained 122 sample types, i.e. 122 different patterns of pretreatment and tests. The most frequent type contains 87 batch entries, 56 types occurred only once. Although well defined sample types consisting of fixed sets of tests can clearly be distinguished, this classification is not yet satisfactory.

In a subsequent type reduction the same alphanumerical properties were considered

- (1) sample pretreatment must be equal for $MS = 1$, otherwise $MS = 0$
- (2) tests—overlap of alphanumerical histograms is calculated with an intermediate type of resemblance requested (see Table 2)

Meta-type nr. 3

Contents:

```

1 1314 CDG_CDG_FBG_CRS_AS_HG_ZMG
2 1314 CDG_CDG_FBG_AS_HL_ZMG
3 1314 CDG_FBG_HG_ZMG
4 1314 CDG_FBG_CRS_AS_HG_ZMG_VLA
5 1314 CDG_CDG_FBG_CRS_AS_HG_HG
6 1314 FBG_VLA CDG_CDG_FBG_CRS_A

```

PROPERTY client code

(144 measurements)

```

143|*****|2
1|*****|13

```

PROPERTY sample type

(144 measurements) Only one value ---> 3

PROPERTY pre-treatment

(144 measurements) Only one value ---> 13-14

PROPERTY tests

(987 measurements)

```

143|*****|CDG
142|*****|HLS
141|*****|FBG
140|*****|CRS
139|*****|AS
138|*****|BS
137|*****|SW
136|*****|VLA
135|*****|FBG
134|*****|

```

PROPERTY batch-size

MEAN VALUE 5.14 STANDARD DEVIATION 6.72 (144 measurements)
 N = 2 measurements beyond histogram limits

```

3~ 0 50|*****|
4~ 1 50|*****|
1~ 4 50|*****|
7 6 50|*****|
7 8 50|*****|
9 10 50|*****|
1 12 50|*****|
8 14 50|*****|
6 16 50|*****|
0 18 50|*****|
2 20 50|*****|
1 22 50|*****|
2 24 50|*****|
1 26 50|*****|
2 higher

```

PROPERTY delay time

MEAN VALUE 10.69 STANDARD DEVIATION 5.24 (144 measurements)

```

1 1 50|*****|
30 5 50|*****|
52 9 50|*****|
41 13 50|*****|
8 17 50|*****|
0 21 50|*****|
2 25 50|*****|
0 29 50|*****|
0 33 50|*****|
0 37 50|*****|
2 41 50|*****|
0 45 50|*****|
0 49 50|*****|

```

Figure 8 Information present in a meta-type after type-reduction, the text VLA denotes a combination of related tests

All MS^0 values were calculated. Sample types were merged while $MS^0 > 0.40$. Using this limiting value, six meta types were formed and five sample types were not classified. The latter were all infrequent and exceptional sample types: they were combined by hand with one of the meta-types or deleted. Also two meta types were infrequent: four and 14 occurrences respectively. One of these contained a rather rare type of sample pretreatment: the other showed a special pattern of tests. On chemical grounds, however, it was acceptable for modelling purposes to merge these meta types with the remaining three.

The major conclusion to be drawn from the type construction and reduction in this particular case is that for modelling purposes the sample entries can almost completely be described as

- (1) samples without pretreatment
- (2) samples with pretreatment code 13
- (3) samples with pretreatment codes 13 and 14

Most information concerning meta-type 3 is shown in Figure 8. This classification has been discussed with the laboratory management who judged it as very useful.

The next point of interest with respect to modelling is whether sample types from different origins receive different treatment, i.e. have different priorities. In order to investigate this, a new type construction is performed based on the properties sample origin and pretreatment. Comparison of the delay times of the thus found sample types shows that there are differences (Table 4). Indeed, the decision to construct the three sample origins was based on this investigation.

Table 4. Mean delay times specified according to sample origin and sample pretreatment together with numbers of entries and the standard deviation (SD).

Sample origin		Sample pretreatment (code)		
		0	13	13-14
0	delay	4.94		
	SD	4.89		
	entries	293	2	0
1	delay	8.61		
	SD	6.85		
	entries	93	4	1
2	delay	10.04	7.20	10.48
	SD	4.49	6.15	4.64
	entries	109	148	143

The results of this analysis with FLA are used to construct a simulation model of the laboratory. Several other data are necessary which cannot be obtained from the historical laboratory data, e.g. measurement times for the tests, working times of the technicians. These data are estimated by interviewing the laboratory staff.

CONCLUSIONS

The SIMULIA implementation of FEA has been shown to be a useful tool for analysis of historical laboratory data. The processing times of the program parts are acceptable in the range 10-20 CPU seconds for the various parts. Not all features of FEA needed to be employed in the sample application. However, these have been tested with data files from former simulation projects.

Although the discussed sample application is a part of a simulation project, it clearly shows how FEA can be used also for non simulation purposes. It makes possible a representation of historical laboratory data in an informative way. In order to do so, the sample entries are classified into a small number of clusters or sample types. Because of the fuzzy set approach, the program does not need a restrictive model of the data. The user is allowed to specify at will which properties are considered for the classification and how strictly they are applied. He can introduce simplifications of the data which are based on his knowledge or experience.

To some extent, the path leading to an optimal representation of the historical laboratory data must be found by trial and error. On several occasions, decisions must be made as to whether a simplification is acceptable or which program settings are likely to yield the best results. Because such a decision may prove wrong, the user is able to correct the mistake by undoing the last step and trying again.

ACKNOWLEDGMENT

The authors thank Ir. L. van Naerssen of NV Regionale Energiebedrijf Dordrecht (The Netherlands) for providing the laboratory data and for his cooperation in this research.

REFERENCES

- 1 R. D. McDowall (ed.) *Laboratory Information Management Systems*, Sigma Press, Wilmslow (1987).
- 2 W. D. Shults and G. A. Gibbon, *Trends Anal. Chem.* **5**(10) IV (1986).
- 3 P. Winkel, *Chin. Chem.* **30**, 1758 (1984).
- 4 T. A. H. M. Janse and G. Kateman, *Anal. Chim. Acta* **159**, 181 (1984).
- 5 B. van de Wijngaert, J. Takeman, J. Klaessens, B. Vandeginste and G. Kateman, *Anal. Chim. Acta* **184**, 151 (1986).
- 6 J. Klaessens, T. Sarrs, B. Vandeginste and G. Kateman, *J. Chemometrics* **2**, 49 (1988).
- 7 T. A. H. M. Janse, The optimal utilization of laboratories for water quality analyses, *Ph.D. Thesis*, University of Nijmegen (1986).
- 8 L. A. Zadeh, *Info. Control* **8**, 338 (1964).
- 9 T. Blaffert, *Anal. Chim. Acta* **161**, 135 (1984).
- 10 H. Bandemer and M. Otto, *Mikrochim. Acta* **11**, 93 (1986).
- 11 L. A. Zadeh, *Fuzzy Sets Syst.* **11**, 199 (1983).
- 12 A. C. Zimmer, in *Decision Making under Uncertainty*, ed. by R. W. Scholz, p. 159, Elsevier, Amsterdam (1983).
- 13 R. Wilemsky, *Lispcraft*, Norton & Co., New York (1984).
- 14 G. M. Birthwistle, O. J. Dahl, B. Myhrhaug and K. Nygaard, *Simula Begin*, Petrocelli Charter, New York (1973).
- 15 N. van Buggenum, J. Klaessens, B. Vandeginste and G. Kateman, *Trends Anal. Chem.* **6**, 81 (1987).

EXPERT SYSTEM FOR KNOWLEDGE-BASED MODELLING OF ANALYTICAL LABORATORIES AS A TOOL FOR LABORATORY MANAGEMENT

JO KLAESSENS, THIO SARIS, BERNARD VANDLINGSTE AND GERRIT KATMAN

Laboratorium voor Analytische Chemie, Katholieke Universiteit Nijmegen, Toernooiveld 6525 ED Nijmegen, The Netherlands

SUMMARY

An expert system (LABGIN) is presented for decision support in analytical laboratories by means of digital simulation. In an interactive manner LABGIN constructs simulation models of laboratory organizations. It makes use of a database of model fragments and applies rules in order to prevent the user providing redundant information and to prevent inconsistent models being constructed. The models are written in the dedicated simulation language SIMULA. After compilation they can be used for simulation experiments. LABGIN can be applied to a wide range of laboratory organizations. An example of the application of LABGIN is presented.

KEY WORDS Laboratory organization Expert systems Digital simulation
Decision support

INTRODUCTION

Laboratory management is concerned with the control of costs per analysis and the optimization of the flow of analytical information from the laboratory to the users. Frequently problems arise which require a thorough investigation of the supply of analytical work as compared with the available means, i.e. instrumentation and manpower. Typical questions are: 'What organizational adjustments must be made in order to process an increasing inflow of urgent samples?' 'Is it profitable to purchase some new and highly expensive apparatus to process a specific group of samples?' Or should these samples be analyzed by a specialized commercial laboratory?

Decision making in analytical laboratories is difficult because all aspects of the organization and the input of samples have to be taken into account. In most laboratories the management can obtain sufficient data on the laboratory organization. Because of the evolving application of laboratory information management systems (LIMS) even more data are available. However, these are unrefined data and provide no direct information to the management.¹ The requested information can be obtained by a simulation model of the laboratory, based on the collected unrefined data. This technique is a powerful means to provide the requested information because it enables the laboratory manager to evaluate the effects of decisions before they are actually carried out.

0886-9383/88/010049-17\$08.50
© 1988 by John Wiley & Sons, Ltd

*Received 16 April 1987
Revised 30 June 1987*

Figure 1. Scheme of a simulation project.

To date several successful applications of digital simulation to analytical^{2,4} and clinical laboratories⁷⁻⁹ have been reported. Three phases can be discerned in a simulation project. First, a simulation model of the laboratory is developed which consists of a model of the sample input and a model of the organization with all relevant objects and relations between these objects. Second, the simulation model is subjected to a test procedure to ensure that the model resembles the real laboratory close enough. The third phase is to use the model for decision support by performing simulation experiments. A possible managerial decision is introduced into the model and evaluated by measuring its effect on the performance of the laboratory.

In order to carry out a simulation project expertise in simulation is needed, which is usually not available within the laboratory. Therefore the problem formulation and the laboratory data are handed over to a simulation expert who carries out the project (Figure 1). During any phase considerable communication between expert and management is necessary, partly because a simulation model must be highly specific. As a consequence it is very time consuming and expensive to carry out a simulation project.

The applicability of simulation as a management model tool will be greatly increased when this major drawback can be overcome. A promising strategy to reach this goal is to replace the simulation expert by an expert system or—as it is also called: a knowledge-based system. Supported by suitable software tools, a member of the staff can carry out a preliminary refinement of the data stored in the LIMS, combine these results with his knowledge about the laboratory organization into a simulation model and carry out the simulation experiments. Apart from being more cost-effective, another advantage is that the model can easily be kept up to date for repeated decision support. The software tools should consist of two systems:

- a system for refinement of historical data in order to obtain the quantitative aspects of the model, such as the number of sample types and the statistical characteristics of the sample input
- a system for automated model construction

In this paper we will discuss the system for automated model construction called LABGEN.

Introductory papers on expert systems have already been published in the chemical literature¹⁰⁻¹². Recently, a few research reports on expert systems for the development of simulation models have been published: MOSES¹³ and the ACD system¹⁴ for models of continuous flow systems and the discrete event simulator of the KFE system¹⁵.

SYSTEMATIC MODELING OF ANALYTICAL LABORATORIES

The dedicated simulation language SIMULA¹⁶ is very suitable for the simulation of systems consisting of classes of objects. A class description can be defined containing all relevant actions and attributes. A specific object refers to the class description and obtains its properties because values are assigned to its attributes. A simulation model of an analytical laboratory contains the class descriptions of the following classes:

sample

sample generator which simulates the sample input flow of the laboratory.

planner which controls the sample flow within the laboratory

analyst,

instrument

These classes are called lab components. The model also contains several queues to store the samples between two actions. In another part of the model the specific objects are defined and the values are assigned to the attributes. Finally, there is a structure which controls the simulation run. A more detailed discussion of the modelling of analytical laboratories with SIMULA has been published elsewhere.¹⁷

The flow of samples through the laboratory is determined by many rules and relations between lab components. An example of a relation is that an analyst can only operate some of the available instruments. Very important are the rules that determine the order of sample

Figure 2. Two examples of laboratory organizations. (A) with centralized planning. (B) with decentralized planning. In boxes: lab components; in ellipses: queues for sample storage; full arrows: sample flow; broken arrows: interaction.

PLANNER (model A)	PLANNER (model B)
-accept entering samples (ACCEPT)	-accept entering samples (ACCEPT)
-[combine samples for batch processing (COMBINE)]	-assign samples to appropriate workstation (TO_STATION)
-store samples in lab.queue (STORE)	
-[keep lab.queue in order (ORDER_QUEUE)]	
-determine workload (ASSIGN_SAM)	
-[schedule urgent samples for immediate analysis (TO_STATION)]	

Figure 3. Basic actions of models A and B of Figure 2. In square brackets: basic actions which are optional within specified organization; in parentheses: corresponding skeleton controls (see text).

processing. These rules may be based on planned due times or different priorities for the samples; the concept of urgent samples may be applied or there may be no rules at all.

There are also differences between laboratories with respect to scheduling of samples to be analyzed. Figure 2 shows two very different laboratory organizations. In model A, every morning the planner determines the workload of that day. It is an example of centralized planning based on the occupation of the instruments. In model B, on the other hand, an entering sample is immediately assigned to one of the analysts. It is the analysts who decide in which order to samples are analyzed. Model B is an example of decentralized planning. All these aspects must be incorporated into the simulation model.

Although the same lab components are present in both models of Figure 2, they clearly do not perform the same actions. Figure 3 shows the various actions that are performed by the planner in models A and B. These actions are called basic actions. Most basic actions for the planner are listed in Figure 3. Each lab component can be designed by combining a number of defined basic actions. However, there are several restrictions with respect to the combinations. First, not all combinations of basic actions are allowed. The basic action 'keep lab.queue in order' for the planner of model B is not allowed because this queue is not present in this model. Second, there are interactions between lab components. The structure of the planner affects the analyst with respect to the required basic actions, as can be concluded from Figure 2. Because also a basic action itself may be affected, alternative versions for one basic action must be defined.

LABGEN FOR AUTOMATED MODELLING

LABGEN's task is to construct the lab components by combining the appropriate basic actions and to incorporate the lab components into the simulation model. This is accomplished in an interactive way by asking questions and transforming the answers provided by the user into parts of the simulation model. For practical reasons LABGEN is written in SIMULA. It is

Figure 4 General view of LABGEN and the lines of communication

implemented on a NAS 9600 mainframe computer. The construction of a complete laboratory model requires less than 3 CPU seconds run time.

Figure 4 gives an overview of LABGEN. Although LABGEN has an inference engine to infer facts with rules, it is not organized as a fully rule-based production system. If the basic actions of a lab component are known, no uncertainty is left about their order. Therefore it would be unnecessarily complicated to have production rules to infer it. LABGEN has a main body in which the actions are carried out in fixed order. All communication with the user is controlled by a flexible central user interface. There are flexible backup facilities to introduce changes in a simulation model. The most important aspects of LABGEN are discussed in detail in the next sections.

The database and refinement and skeletons

LABGEN makes use of a database containing the SIMULA code of the different versions of the basic actions. These are referred to as skeletons. The code of the skeletons is not directly executable as this would increase the number of skeletons drastically. Certain parts are left undefined and contain a command for refinement instead. This command is executed by LABGEN. Figure 5 shows two alternative skeletons for the basic action 'assign samples to appropriate workstation'. The first two positions of each line contain information to be used

[illegible]

Figure 5. Two alternative skeleton FO-STATION. The first two positions contain information to be used by LABGEN. A command for refinement starts at the second position.

by LABGEN. A command for refinement is recognized because it starts at the second position. When such a command is met, the appropriate procedure fires which introduces the refinement into the model.

For each basic action a frame-like structure called skeleton control is defined (Figure 6). These skeleton controls contain the procedures for skeleton refinement. They also contain information to select the appropriate version of the skeleton and to locate it in the database. All skeleton controls share a program structure at a hierarchically higher level for the communication with the database.

If a basic action is present in the model the corresponding skeleton control is activated by the supervisor of the lab component. LABGEN has five supervisors for the different lab components (Figure 6). They contain knowledge about the basic actions that may be present in the lab component. Data to quantify the attributes of the lab component are stored in the supervisor. This aspect will be discussed later.

The supervisors are activated in sequence. They are the only parts of LABGEN that are executed in a fixed order. The procedures for skeleton refinement are completely controlled by the skeleton under consideration. Because of this flexible and very modular structure,

Figure 6 The supervisors for each lab component and their skeleton-controls

introduction of a new alternative skeleton is easy. Only procedures for refinement and knowledge to select this alternative need to be added to LABGEN.

Facts and rules

During any stage of the construction of a simulation model decisions are made which are ultimately based on the answers of the user. In order to allow communication between the various parts of LABGEN an intermediate structure between an answer of the user and the application of this information is developed. All decisions are based on facts which are instantiations of the frame FACT (Figure 7) and can be considered as units of information.

Evaluation of a fact is carried out by an inference engine. If the fact is already determined then its value is returned, otherwise the value is determined first. The inference engine tries to infer this value with rules using a depth first backward reasoning process. If this is not successful the procedure ASK_USER of the fact is executed. Once a fact has been evaluated this information is available in any part of LABGEN. An order to evaluate a fact is given at the level of the skeleton-controls. As a consequence facts which are not relevant to the model

Figure 7 Structure of facts and rules in LABGEN

at hand are never evaluated and the corresponding questions are never asked

LABGEN contains about 80 facts which define all structural aspects of a simulation model. Some examples of facts and their meaning are

only_duet planning of sample scheduling is based on the due time of the sample
 plan_centr the daily workload is determined by the planner
 change_prio the parameter used for sample scheduling changes in time
 order_queue the lab_queue (see Figure 2) must be re ordered every day

The rule base contains about 70 rules. They are also defined as a frame (Figure 7). The value of several facts is dependent on the values of other facts. The knowledge about these relations is stated by the rules e.g. the knowledge to rule out a specific combination of basic actions. Two examples of rules containing the previously mentioned facts are

IF only_duet = TRUE OR plan_centr = FALSE
 THEN change_prio = FALSE

IF plan_centr = TRUE AND change_prio = TRUE
 THEN order_queue = TRUE

The first rule states that the due time of a sample does not change in time and that in the case of decentralized planning a time dependent planning parameter is not allowed, as this would cause an inconsistency in the model. The second rule states that the fact order_queue is true if lab_queue is present in the model (only in case of centralized planning) and the order of the samples in this queue is time-dependent.

LABGEN also has a mechanism for forward reasoning which uses the same rules. This belongs to the part of LABGEN that controls the backup and will be discussed below.

The use of facts and rules in LABGEN differs from fully rule-based production systems. In the latter the manipulation of facts and rules is the basic mechanism to infer the value of a goal conclusion. LABGEN's basic mechanism consists of the interactions between the supervisors and skeleton-controls and between skeleton controls and skeletons in the database. Facts serve as a tool for communication between the various parts of LABGEN. Rules are applied to incorporate knowledge in order to prevent the user providing information which is actually redundant and to prevent the construction of inconsistent simulation models.

Attributes of the lab components

During a run of LABGEN the attributes of the lab components and their constraints are dynamically determined because they are dependent on the basic actions that are present. Each time a new attribute becomes necessary a new instantiation of the frame INPUT-ATTRIBUTE is defined (Figure 8). All constraints of the attribute are stored in it: the name of the attribute, its type (whether it is a real, integer or text), dimensions (single value, array or matrix), the range of the allowed values, etc. The new INPUT-ATTRIBUTE is stored in the VARIABLE-QUEUE which belongs to a program structure called VARIABLE-SET (Figure 9). This structure is available in the supervisors of all lab components.

After the structure of the simulation model has been determined the attributes of the objects of the lab components are quantified. All necessary information to control this is available in the VARIABLE-SET of the supervisor. Each object in the simulation model has

Figure 8 The frame INPUT ATTRIBUTE

its own VARIABLE_SUBSET. When an object is being quantified every INPUT ATTRIBUTE in the corresponding VARIABLE_QUEUE is considered one after another. First it is checked whether the data are read in from the backup file (see next section) and subsequently stored in the DATA_QUEUE. If they are not, the procedure ASK_USER of the INPUT_ATTRIBUTE is executed. The input data are rigidly controlled by the procedure TEST_VAL, e.g. whether they are in the allowed range. If necessary, corrective action is undertaken. If the data meet the constraints, they are incorporated into the simulation model and also stored in the INPUT_ATTRIBUTE itself.

By use of the program structure VARIABLE_SET the input of quantitative data has been separated from the input of qualitative data concerning the structure of the model. At the same time, this structure allows control of the input of data. Input of wrong data—e.g. a real instead of an integer value—would cause run time errors in the simulation model, which, of course, must be avoided for LABGEN to become of practical value. The rather complex structure of VARIABLE_SET is needed in order to allow a flexible use of the backup.

The backup facilities

The facts that are determined and the INPUT_ATTRIBUTES that are defined contain all the information that is necessary to reconstruct a simulation model. At the end of a session with LABGEN, this information is always stored in a backup file. If a backup file is present, a session always starts with reading this file. The appropriate facts are set, determined and obtain the stored value. The quantitative data are put in the DATA_QUEUE of the

Figure 9 The program structure VARIABLE SET which is present in the supervisors of all Lab components

Figure 10 Activities with respect to the backup specified according to the three options

appropriate `VARIABLE_SUBSET` (Figure 9). The use of the backup does not cause any differences to the run of LABGEN's main body, i.e. the reading and refinement of skeletons, except that parts of the required information are not obtained from the user but from the backup file instead.

There are three options with respect to the backup (Figure 10):

- 1 rerun LABGEN,
- 2 manipulate quantitative data,
- 3 change properties of lab components

Option 1 is not very complicated. The program is run again and information which is not present in the backup file is requested from the user. This option can be used to continue a session which has been interrupted.

During the development of a simulation model, it frequently occurs that data are necessary which can only be guessed. A first guess usually has to be tuned during a subsequent validation stage. Therefore it is important that quantitative data can be easily changed. For this purpose option 2 may be chosen, which allows any attribute of any object to be changed. The program runs through the `DATA_QUILUF` of the appropriate `VARIABLE_SUBSET` and any data unit may be changed. At a later stage these are moved to the right `INPUT_ATTRIBUTES`, as has been mentioned in the previous section. At that time the data are controlled and, if needed, corrective action is undertaken.

Option 3 allows any structural aspect of a simulation model to be changed, including the number of objects of a lab component. If the number of objects is decreased, the user must indicate which specific objects are to be removed. The corresponding `VARIABLE_SUBSETS` are deleted. If an object is added, the necessary data are entered by the mechanism discussed in the previous section. A change of structural aspects is accomplished by allowing facts to be changed.

If a fact has been changed, a rather complicated mechanism is activated. As has been pointed out before, some facts can be inferred by means of rules. Changing one fact may affect other facts whose values are obtained from the backup and belong to the previous run. In order to cope with this problem, LABGEN applies a forward-reasoning process using the same rules. This means that, if any of the facts of the precondition of a rule has been changed, the fact of its conclusion is set undetermined. Subsequently it is determined again by the

backward reasoning process described earlier. In this way, any effect of one fact on another is accounted for. Changing one fact may require several other questions to be answered.

Facts may also affect the presence of attributes of lab components and their constraints. Therefore, LABGEN has a special class of rules with a precondition of one fact and a conclusion with the identities of one or more INPUT ATTRIBUTES. If the fact of the precondition has been changed, the INPUT ATTRIBUTES of the conclusion are removed from any VARIABLE-SUBSET (Figure 9). During the run of the main body of LABGEN, new INPUT ATTRIBUTES are defined for the attributes that have now become relevant and are added to the appropriate VARIABLE-QUEUES.

Although all information concerning the simulation model is stored in the backup file, it is not easily readable by the user. Therefore, the information is also stored more clearly arranged in a so-called session file.

APPLICATION OF LABGEN: AN EXAMPLE

A session with LABGEN results in a simulation model consisting of 800-1500 lines of SIMULA code. After compilation, the model can be run. A simulation model may be constructed in two modes: the preview mode and the production mode. In the preview mode, all actions during the simulation are printed. This mode allows the user to check whether the sequence of actions within the laboratory is sensible. The production mode is suitable for simulating a longer period of time, because the individual actions are not written out.

LABGEN is intended to cover a wide range of laboratory organizations. As a consequence, many structural aspects of the laboratory have to be specified. However, some major classes of models can be discerned:

- a1 Continuous sample input and scheduling: the samples may enter at any time of the day and are immediately scheduled. This class of models is suitable for laboratories in which many samples must be analyzed within a few hours (small delay times), e.g., clinical laboratories.
- a2 Discontinuous sample input and scheduling: sample entrance and scheduling is performed only once a day. This class of models is suitable for laboratories in which delay times of several days are usual, e.g., industrial laboratories for quality control. Because of the larger delay times, it is not relevant to consider the exact time of a sample entry.
- b1 Centralized planning: the workload of a day is planned centrally by the planner (see Figure 2).
- b2 Decentralized planning: the analysts determine which samples are analyzed. This organization, of course, is more flexible but is more difficult to control.

To some extent, LABGEN itself decides which class of model is most suitable for a particular laboratory.

Several other aspects of the application of LABGEN will be illustrated with an example of a fictitious simulation project in an analytical laboratory. In this laboratory, for quality control, samples are analysed on two AAS instruments and two HPLC instruments. There are six different sample types: three are analysed on AAS and three on HPLC. The laboratory is staffed by two analysts and on busy days a third analyst is called in. The analysts can operate all instruments. Because of changes in the production process, it becomes desirable to reduce the delay times of the samples, i.e., the time lag between the entrance of the sample and the time the analytical result becomes available.

```

number of sample types --> 6
number of analysts --> 3
number of sample generators --> 1
number of instruments --> 4
The laboratory is able to process urgent samples?
No
Dvertime of all samples is always larger than 1 day?
Yes
The week-ends are discarded for time calculation during
the simulation run?
Yes
Samples can enter the laboratory in batches?
No
Are there any sample types with more than 1
analysis requested?
No
The priority of a sample is determined by its
planned due time only?
Yes
An analyst can operate all instruments?
Yes
The planner assigns samples to an analyst?
if 'no' then assignment to an instrument
No
The actual analysis jobs are planned by the planner?
Yes
Samples can be analysed in batches?
No
Instrument breakdown is allowed in the model?
Yes
Name JOHN
Begin times for each week day
Value 6 8 8 8 8 0 0
Working times for each week day
Value 6 8 8 8 8 0 0
Minimum number of jobs to be exceeded for reporting
Value 3
How many days before first activation?
Value 3

```

Figure 11 Parts of the session file which is constructed after a session with LABGEN and which contains information about the simulation model

After a preliminary refinement of the historical data of the laboratory, a simulation model of the laboratory is constructed in a dialogue with LABGEN. Figure 11 shows parts of the session file which is constructed after a session with LABGEN. It gives a good impression of the dialogue between user and computer. The qualitative data to specify the objects of a lab component are entered during the same session. Figure 11 also contains the qualitative data concerning one of the analysts.

The next step is to check the validity of the simulation model. It is very useful to examine the actions performed in the simulated laboratory. Figure 12 shows day 44 of a simulation run in

simulation day 44, weekday 2

daytime (h)	gen	type					
		1	2	3	4	5	6
0 02	1	1	1	2	2	3	

day- time (h)	anal			inst			
	1	2	3	1	2	3	4
8 00	R1	R3					
8 50	*R1**						
8 57		*R3**					
8 59	*R4**						
9 20		*R6**					
9 66	*R1**						
9 90		*R3**					
10 20		*R4**					
10 23	*R5**						
10 75	*A1**						
10 79		*A5**					
11 15				A1			
11 28		*A4**				A2	
11 57	*A1**			*****			
11 76		*A3**				A2	
12 07				A1			
12 36					A2		
12 48	*A1**			*****			
12 80				A1			
12 84						*****	
12 98		*A6**			*****		
13 22	*A3**			*****			
13 41		*A2**				A2	
13 80					A1		
13 90				A2			
14 35		*A2**		*****			
14 35	*****				*****		
14 53					*****		
14 60	A4						
14 79				A2			
14 91	*****					A1	
15 10		*A2**		*****			
15 30						*****	
15 41	A5						
15 66				A2			
15 85	*****					A1	
15 97		*****		*****			
17 48						*****	
17 79						*****	

Figure 13 The actions in the laboratory during day 44 of a simulation run of the model in preview mode generated by LABGEN

the preview mode. First the sample entries of this day are shown. Second the actions of the analysts are shown. The third analyst is not present. At 08:00 the two analysts start with a report job of a sample which has been analysed the previous day (denoted by R, the number refers to the sample type). A report job consists of calculating the analytical result and writing a report. At 08:50 analyst 1 completes his first job. The completion of a job is denoted by ***** or a letter and a number surrounded by * and ** if the analyst immediately continues with another job. At 10:75 analyst 1 starts with an analysis job (denoted by A, the number refers to the sample type). At 11:15 he has finished the sample preparation and activates instrument 1 (AAS) for the measurements. The activation of an instrument is also denoted by 'A', the number however refers to the analyst who operates it. During the measurements both analyst 1 and instrument 1 are occupied. At 11:28 analyst 2 finishes the

Delay times of all samples

Mean value 4.69, standard deviation 4.68 (2951 measurements)
 4.8 % measurements beyond histogram limits

Figure 1. Example of a histogram recorded during a one year simulation run of the model generated by LABGEN.

sample preparation of a sample of type 5 and activates instrument 4 (HPLC). During the measurements of this instrument, the analyst is free to carry out other jobs (LABGEN allows several types of interaction between analyst and instrument).

During the simulation run, several performance parameters are measured and stored in histograms. Figure 13 shows the histogram of delay times of all sample types recorded during a one year simulation run of the model. These parameters can be used for validation of the model. Among others, the following questions are considered for this purpose:

Are the daily effective working times of the analysts in the model reasonable?

Is the extra analyst employed at a reasonable level?

Does the distribution of sample input of the simulation model differ significantly from the real laboratory?

Does the distribution of sample delay times of the model differ significantly from the real laboratory?

The validated model can be used for decision support.

The first point of investigation is whether the laboratory can meet the request of shorter delay times without any changes in organization. In order to investigate this, the planned due time of each sample type in the model is adapted to the new situation (Table 1) and the delay times are measured during a simulation run of one year. Table 2 shows the results of this run as compared with the old situation. Although the laboratory is flexible enough to allow a reduction in delay times, some sample types cause problems, especially type 4 which is analysed on instrument 3 (HPLC). Because AAS samples (types 1-3) give no problems and the effective working times of the analysts are not affected by the new situation, it is concluded that the HPLC instrument is the bottle neck. There are two possible solutions to this problem:

more HPLC equipment

introduction of auto-samplers which allows batch processing of samples

Table 1 The planned due time for each sample type in the simulation model for the old and the new situation

Sample type	Planned due time in model (days)	
	Old situation	New situation
1	2	2
2	9	4
3	4	2
4	8	4
5	8	4
6	8	4

Table 2 Performance of the laboratory in the old and the new situation (without changes in organization) mean delay time of the sample types and the percentage overdue i.e. the percentage of samples not analysed within the planned due time

Sample type	Delay time (days)			
	Old situation		New situation	
	Mean	% overdue	Mean	% overdue
1	1.0	0	1.2	0
2	1.9	0	1.7	0
3	1.4	0	1.3	0
4	12.2	79	8.3	82
5	6.6	11	3.7	24
6	6.7	13	3.8	26

In this example the second possibility is investigated. HPLC samples of the same type may be combined into a batch (90% chance). The samples of a batch are analysed together, which allows reductions in preparation time, analysis time and report time of combined samples. These changes are introduced into the simulation model in a dialogue with LABGEN as shown in Figure 14. The results of a one year simulation run of the revised model are shown in Table 3. Clearly, the delay times of the various sample types are much more balanced. At the same time, the extra analyst appears to be needed only during four days, instead of 117 days in the old situation. The advice from this simulation project would be the application of auto samplers to allow batch processing of analyses.

This simulated example is intended to illustrate the application of LABGEN to solve a specific problem in an analytical laboratory. Several important aspects of simulation are not mentioned, such as designing the experiments to be conducted with the model. These are discussed in more detail elsewhere.⁶

```

Data are read from the backup file.
Possible options to continue:
a) Rerun the program with data read from backup file
b) Manipulate data of objects of lab components
c) Change properties of simulation model
c

You can remove the following property
  Default is ----> No do not remove
  To select default ----> press ENTER
Samples can be analysed in batches?
  The previous answer given -->No
y

Samples can be analysed in batches?
y

Some types of samples are only analysed if some
minimum batchsize is exceeded?
n

Should the actual combination of two samples into
one batch be dependent of a combination chance,
if all other constraints are met?
y

Combination chance for each sample type
REAL array 1 to 6
restrictions: 0.00 < input-value < 1.00
enter EXACTLY 6 values
0 0 0 9 9 9

Reduction factor in preparation times because of batch combination
REAL VALUE
restrictions 0.01 < input-value < 1.10
0 4

Reduction factor in analysis times because of batch combination
REAL VALUE
restrictions 0.01 < input-value < 1.10
0 6

Reduction factor in report times because of batch combination
REAL VALUE
restrictions 0.01 < input-value < 1.10
0 8

```

Figure 14 Dialogue with LABGEN to introduce batch processing into the simulation model

Table 3. Performance of the laboratory with batch combination: mean delay time of the sample types and the percentage overdue (see Table 2)

Sample type	Delay time (days)	
	Mean	% overdue
1	1.4	3
2	2.6	1
3	1.5	6
4	2.6	6
5	2.4	4
6	2.4	3

CONCLUSIONS AND FUTURE RESEARCH

The implementation of LABGEN is completed. In essence, LABGEN appears to be flexible enough to generate simulation models for a wide range of laboratory organizations. The next step of research is the application of LABGEN to real laboratory organizations to investigate whether the scope of structural concepts is wide enough to cover real situations. This is likely to result in some refinements and the addition of some new concepts.

Because of its class concept, SIMULA is very suitable for constructing expert systems with a rule base of moderate size. LABGEN is not organized as a fully rule-based system. Its control structure is much too specialized to be generalized into an empty shell. However, its approach of embedding a rule manipulation system in conventional program structures is of interest for other domains too, especially domains which require considerable computation, such as statistical data analysis. SIMULA is much more suitable for calculations than the AI languages LISP and PROLOG or empty shells.

REFERENCES

1. W. D. Shults and G. A. Gibbon, *Trends Anal. Chem.* **5**(10) IV (1986).
2. B. G. M. Vandeginste, *Anal. Chim. Acta* **112**, 253 (1979).
3. B. G. M. Vandeginste, *Anal. Chim. Acta* **122**, 435 (1980).
4. J. G. Volkenbroek and B. G. M. Vandeginste, *Anal. Chim. Acta* **133**, 85 (1981).
5. I. A. H. M. Jansz and G. Kateman, *Anal. Chim. Acta* **159**, 151 (1984).
6. B. van de Wijngaert, J. Lakeman, J. Klaessens, B. Vandeginste and G. Kateman, *Anal. Chim. Acta* **184**, 151 (1986).
7. I. K. Vaananen, S. Kivittikko, J. Koskenniemi and A. Rindler, *Methods Inf. Med.* **13**, 158 (1974).
8. B. Sandblad, in *Proc. Int. Symp. and Course Simulation 1975*, ed. by M. H. Hamza, p. 364, Acta Press, Calgary, Zurich (1975).
9. P. Winkel, *Chim. Chem.* **30**, 1758 (1984).
10. R. I. Dossy (ed.), *Anal. Chem.* **56**, 1200A, 1312A (1984).
11. J. W. A. Klaessens, G. Kateman and B. G. M. Vandeginste, *Trends Anal. Chem.* **4**, 114 (1985).
12. J. Klaessens and G. Kateman, *Fres. Z. Anal. Chem.* **326**, 203 (1987).
13. J. Rütt, *Informatik Fachberichte* **85**, p. 505, Springer, Berlin Heidelberg New York Tokyo (1984).
14. G. I. Doudikis and R. J. Paul, *J. Oper. Res. Soc.* **36**, 319 (1985).
15. *The Knowledge Engineering Environment*, Intelliparc, Menlo Park (1984).
16. G. M. Burthwhistle, O. J. Dahl, B. Myhrhaug and K. Nygaard, *SIMULA Begin*, Petrocelli Charter, New York (1973).
17. N. van Buggenum, J. Klaessens and G. Kateman, *Trends Anal. Chem.* **6**, 51 (1987).

LABORATORY ORGANIZATION AND MODELLING ASPECTS*

SUMMARY

An expert system for management support of analytical laboratories by means of simulation is presented: LABGEN. In an interactive session with the user it constructs a simulation model of the laboratory without any programming or knowledge about simulation being required. The paper discusses the organizational aspects of the laboratory and how they are modelled by LABGEN.

APPLICATION OF DIGITAL SIMULATION FOR MANAGEMENT SUPPORT

Optimization of the amount of the information provided by an analytical laboratory as compared to the corresponding costs per analysis is one of the major objectives of the laboratory management. Because the value and the costs also depend on the organization of the laboratory [1], decision making about the organization becomes of critical importance. This is complicated because many sources of uncertainty are involved, e.g. fluctuations in the sample input stream and in the available manpower in the laboratory, human factors such as man-machine interrelations.

Many historical data about the laboratory are available, e.g. stored in a LIMS. The data can be used to support the management during the

* This chapter is published as: J. Klaessens, L. van Beysterveldt, T. Saris, B. Vandeginste, G. Kateman, LABGEN, expert system for knowledge based modelling of analytical laboratories. Part I: Laboratory organization, Anal. Chim. Acta, submitted for publication

decision making process [2]. A method to do this is by applying digital simulation. This method implies that a computer or simulation model of the laboratory is constructed and validated [3,4]. The model should resemble the real laboratory in its statistical behaviour. After validation, the model can be used for decision support: a possible decision (e.g. employment of an additional technician) is incorporated into the simulation model and evaluated by measuring its effect on the performance in a number of simulation runs. If the effect is satisfactory, the decision can be applied to the real laboratory.

Although the value of simulation for decision support in analytical laboratories has been demonstrated in a number of applications [5,6,7], a widespread use of the method has been hindered by two serious drawbacks: a simulation project was expensive and time consuming. These drawbacks are mainly caused by two factors. First, programming of a simulation model requires a simulation expert. Second, a simulation model must be very specific and many data about the laboratory are needed. As a result of this, a considerable amount of communication between simulation expert and laboratory staff was necessary. This situation can be compared to the development of an expert system by a knowledge engineer.

In order to overcome these drawbacks, two software tools have been developed which allow the laboratory staff to carry out a simulation project in a short time and without any programming needed. The first software tool is the program FEA (Front End Analysis) for analysis of historical laboratory data in order to obtain the quantitative aspects of the simulation model [8]. The second tool is an expert system LABGEN for automatic construction of simulation models of analytical laboratories [9].

AUTOMATIC MODELLING BY LABGEN

LABGEN is a generator of simulation models of analytical laboratories and employs an expert system shell (called INFER) for control of the generation process. The application of expert system techniques is becoming increasingly important [10-13]. Figure 1 shows the structure of LABGEN. It is an interactive program which requests all relevant information from the user: qualitative and quantitative data about the structure of the laboratory. LABGEN makes use of a database of model fragments called skeletons. They consist of blocks of unrefined computer code in the language SIMULA [14]. LABGEN picks up the proper skeleton from the database, refines it by employing a specific program structure skeleton-control, and incorporates it into the simulation model. LABGEN must make sure that this part of the model does not interfere with any other part of the model which is already defined. In order to do so, the depth-first backward reasoning mechanism of INFER is used.

LABGEN has facilities to provide additional explanation for any question if desired. Such an explanation is read from a help file. LABGEN does not give information about the reasoning process itself (so-called Why? question). This is considered not to be very informative because a run of LABGEN is not one reasoning process to solve a general goal problem but there are many parallel processes which are shallow.

All information provided by the user is stored in a backup file to be used by LABGEN and in a session file which is more suitable to be read by the user. Using the backup file, LABGEN has extensive backup facilities which allow the user to

- stop a session with LABGEN at any time and continue from the same question afterwards,
- store former simulation models,
- easily introduce changes into a model (e.g. for model refinement) while the new model is fully checked for consistency.

The structure of LABGEN has been previously described in detail [9]. In Part I of the present series of papers [15] the simulation and laboratory organization aspects of LABGEN are discussed. In order to do so,

Figure 1 Structure of LABGEN

a very simple simulation model generated by LABGEN is described. Using this as a starting point, more sophisticated simulation features which are necessary for a model to approach reality, are described. In Part II application of LABGEN to model a real laboratory is discussed.

SIMULATION MODEL OF A SIMPLE LABORATORY ORGANIZATION

It is important that a simulation model resembles the real laboratory as well as possible. If the model is too abstract, the laboratory manager will hardly have an opportunity to grasp a view of it, which is necessary in order to have confidence in the model and the predictions that are based on it. Therefore, LABGEN constructs a simulation model by using a number of building blocks called lab components:

- sample;
- sample generator which simulates the sample input flow of the laboratory;
- planner which controls the sample flow within the laboratory;
- technician;
- instrument.

The lab components are modelled with SIMULA's class concept. Each lab component is programmed as of a class of objects, which contains all relevant actions and attributes. In a simulation model several objects of the class "instrument" may be present, each of which can be distinguished from all other instruments. A typical laboratory model contains several objects of the lab components (except the "planner", of which only one object can be present). Apart from these lab components, a model also contains a number of dynamic queues to store the objects of "sample" between two actions.

The structure of a very simple simulation model generated by LABGEN is shown in Figure 2. The generator generates a random number of samples every morning. They are stored in the queue LAB_IN until the planner takes them out and sends them to the appropriate instrument. In this particular example, this is the only activity of the planner and is performed only once a day, after sample entrance. Each instrument has a sample_queue in which the samples are stored until analysis. Instruments are passive objects. For further processing of the samples actions by the technicians are necessary. The technician searches the sample_queues of the instruments he may operate. If a sample is found, the analytical process is carried out after which the sample leaves the system.

Figure 2 Very simple simulation model showing the lab components (in boxes), the dynamic queues (in ellipses), sample flow (in full arrows) and interaction during sample analysis (in broken arrow)

Part of the SIMULA code of this technician is shown in Figure 3. The procedure for finding a suitable job is not included in it for reasons of length. As has been mentioned before, the model is constructed in a dialogue with LABGEN. The session file gives a good impression of this dialogue because it contains the questions and the answers. Figure 4 shows all questions in the session file concerning the technician of the model. As is shown, there are two objects of this class, for each of which numerical data are entered such as working times for the weekdays. The data to quantify a simulation model are very important. Even for a very simple model such as this example, many quantitative data are necessary (Table 1).

```

Process CLASS technician(num,iden);
VALUE iden; TEXT iden; INTEGER num;
BEGIN
    ....
    PROCEDURE do_analysis;
    BEGIN
        inst.avail := FALSE; inst.samp :- samp;
        inst.analt :- THIS technician;
        analyzing := analyzing + samp.prep_time;
        Hold(samp.prep_time);
        IF NOT inst.breakdown THEN
            BEGIN
                analysis_time := 0;
                IF inst.anal occ = 1 THEN
                    analysis_time := samp.anal_time;
                ACTIVATE inst;
                analyzing := analyzing + analysis_time;
                Hold(analysis_time);
                reporting := reporting + samp.rept_time;
                Hold(samp.rept_time);
            END;
            inst :- NONE;
        END---do_analysis---;

        PROCEDURE day_actions;
        BEGIN
            IF Time < 0.01 THEN Hold(act after*24+0.03);
            IF work_times(weekday) < 0.01 OR in_trouble THEN
                GOTO end day;
            Hold(beg_times(weekday)-now);
            analyzing := reporting := 0;
            WHILE now < beg_times(weekday)+work_times(weekday)
            DO BEGIN
                IF NOT find analysis THEN Hold(0.25)
                ELSE do_analysis;
            END;
            h_work.input(analyzing+reporting);
            end day;
            Hold(24-now+0.02);
        END---day_actions---;
        ....

        WHILE TRUE DO day_actions;
    END***technician***;

```

Figure 3 Part of the SIMULA code of the technician of a simple model generated by LABGEN (see Figure 2)

```

Enter the number of technicians (if relevant: including
reserve technicians).
2

Technicians can operate all instruments?
N

The model has one or more reserve technicians which are
employed in times with a high workload?
N

Technicians may be struck by sudden illness in the model?
N

Reporting jobs are gathered?
If no: reporting immediately after analysis.
N

A technician must finish an already started analysis job
at the end of the day?
N

      **** Data ****

Name JOHN

Begin times for each week day
Value : 8 12 8 8 8 0 0
Working times for each week day
Value : 8 4 8 8 8 0 0
Numbers of the instruments operated by this technician;
Value : 1 2

Name PETE

Begin times for each week day
Value : 8 8 8 8 8 0 0
Working times for each week day
Value : 8 8 8 4 8 0 0
Numbers of the instruments operated by this technician;
Value : 3 2

```

Figure 4. The question concerning technician asked during the construction of the model of Figures 2 and 3

Table 1

Attributes of the lab components to be quantified for the simulation model of Figures 2-4.

main	length simulation run
sample types	preparation times
	measurement times
	report times
	instruments needed
	due times (planning)
generator	working days
	sample entries per day
	distribution of sample types
planner	working days
technician	working days and times
	instruments operated

Although this model may be suitable to simulate a specific real laboratory, it will be clear that most laboratories have a structure which is too complex to be modelled in this way. Applicability of LABGEN in practice depends on the way it can handle the complexities of reality. In order to cover these situations, LABGEN can incorporate many additional features into the simulation model. They can be incorporated in any combination as long as they do not exclude one another.

SIMULATION AND LABORATORY ORGANIZATION FEATURES OF LABGEN

The major laboratory organization features which can be modelled by LABGEN are described below. This survey is useful not only to show LABGEN's applicability to a wide range of laboratory organizations. But it is also a list of organization aspects which should be considered by the laboratory manager in order to realize an optimal setting of the laboratory organization.

i) Continuous vs. discontinuous model

In the basic model described above, samples are generated only once a day. This approach is only suitable for laboratories in which large delay times are usual, i.e. delays of several days to several weeks between the entrance of the sample and the reporting of the analytical result to the customer. In this case the exact time of entrance is not important, only the date. This so-called discontinuous model is favourable because it is simpler in structure. These long delay times have been observed in real laboratories [6,7,16], especially laboratories for quality control.

In clinical laboratories or laboratories for process control the throughput is much faster. If the delay times are in the order of magnitude of hours, the exact time of sample entrance becomes relevant in the model. It will make difference whether a sample arrives in the morning or in the afternoon just before the laboratory is closed. The discontinuous model is not appropriate for this type of laboratories. LABGEN can construct a continuous model in which the generator is active during a part of the day. Sample generation then becomes a continuous process. This means that the generator generates a sample entry, waits a so-called interarrival time, and then generates the next sample entry. The interarrival time is a random value according to a specific distribution. Figure 5 shows flow diagrams of both alternative operation modes of the genera-

a

b

Figure 5 Flow diagrams of the operation of the sample generator in a) a discontinuous and b) a continuous model.

tor. Because the samples may arrive at any time as long as the generator is active, also the planner is active during this period in order to immediately transmit the samples to the proper instruments.

ii) *Centralized vs. decentralized planning*

An important activity in a laboratory is the planning of sample scheduling. This means in fact the determination of the order in which the samples are processed. Apart from how the order is determined (this is discussed in the next section), it is also important by whom it is done.

A laboratory as described by the basic model has a decentralized planning. All samples are immediately sent to the instrument to be used. The technician chooses an instrument to operate and takes the sample with the highest priority from its sample_queue.

Figure 6. Simulation model with centralized planning: lab component (in boxes), queues (in ellipses), sample flow (in full arrows), interactions during scheduling and sample analysis (in broken arrows).

An alternative mode of planning is in a centralized way by the planner. Figure 6 shows a model of a laboratory with centralized planning, as opposed to a model with decentralized planning shown in Figure 2. Every morning the planner determines the workload: samples are scheduled until a certain occupation is realized for the instruments or techni-

cians (expressed as numbers of analyses or expected analysis time). Before scheduling the samples are stored in the lab_queue.

Centralized planning is suitable if the processing times of the samples and the delay times both are relatively large. Because of its centralized character, the main advantage is the possibility to better attune the activities of the technicians, especially if there are samples which can be analysed on more than one instrument.

Incorporation of centralized planning into a simulation model adds of course to the complexity of the model. However, it has the additional advantage that the activities of the technicians in the model can be more strictly controlled as it is impossible to process samples which are not yet scheduled in the workload, whatever spare time may be available. Processing of a smaller number of samples than scheduled, on the other hand, can occur if the progress of the work happens to be slow.

iii) Planning of samples in time

In most laboratories a FIFO (First In First Out) order of sample processing is not possible. The throughput of some sample types must be faster, e.g. because the sample can be stored only for a short time or the analytical result is needed sooner. There are two major planning mechanisms to bring about this differentiation in delay times: due times and priority values. Application of due times means that for every entering sample a time is set when (according to the planning) the analysis must be completed. Samples whose due times will be elapsed first (or already are) are selected for analysis first. This mechanism is appealing because it is intuitively clear: sample scheduling is directly related to the time the result is needed.

Employment of priority values probably arises from theoretical studies. Each entering sample obtains a priority value according to its type. Samples with higher priorities are always analysed before samples with lower priorities. In theoretical models this planning mechanism allows calculation of expected values of the delay times [17]. In practical situ-

ations the problem is that the throughput of low priority samples becomes extremely slow as the fraction of high priority samples increases

Table 2

Simulation experiments with the basic model with two instruments and two technicians delay times for four input stream compositions Planning with priority values type1 priority 1 and type2 priority 2 Between parentheses standard deviation of the distribution

ratio type1/type2	delay times (in days)			
	all samples	type1	type2	
0.1/0.9	1.10 (3.66)	8.67 (10.5)	0.48 (0.60)	
0.3/0.7	1.15 (2.51)	3.81 (3.54)	0.12 (0.33)	
0.5/0.5	1.11 (1.84)	2.32 (2.10)	0.02 (0.15)	
0.7/0.3	1.12 (1.43)	1.65 (1.47)	0.00 (0.00)	

The difference between planning with due times and priority values is demonstrated by a number of simulation experiments. Both planning mechanisms are incorporated into the basic model with two technicians and two instruments. There are two sample types having the same characteristics, except with respect to planning. Type1 must be processed within a few days and type2 as soon as possible. This is modelled in both mechanisms as follows

- type1: priority value 1 or due time 2 days;
- type2: priority value 2 or due time 0 days

Table 3

Simulation experiments with the basic model with two instruments and two technicians delay times for four input stream compositions Planning with due times type1 due time = 2 days, type2 due time = 0 days Between parentheses standard deviation of the distribution

ratio type1/type2	delay times (in days)		
	all samples	type1	type2
0.1/0.9	1.21 (1.14)	2.20 (1.30)	1.12 (1.07)
0.3/0.7	1.12 (1.14)	1.92 (1.21)	0.78 (0.91)
0.5/0.5	1.10 (1.14)	1.65 (1.19)	0.59 (0.80)
0.7/0.3	1.12 (1.13)	1.43 (1.14)	0.44 (0.72)

Tables 2 and 3 contain the measured delay times for both mechanisms for four different ratios of the sample types in the input stream. In case of priority values, the delay times show a strong dependency on the composition of the input stream. This effect is very small if due times are applied. The mean delay time of all samples is equal in all measurements because the amount of work in the laboratory is not affected by the changes.

The problem of large delay times for low priority samples can be avoided by defining the priority as a function of time. Depending on the sample type, the priority increases with the waiting time in the laboratory. This mechanism allows very sophisticated planning although some practical problems are associated with it. It requires much calculation in the laboratory and it will be difficult to determine the exact function of time.

These planning mechanisms are provided by LABGEN. In fact, the time dependent priority functions may be used in order to capture plan-

ning rules which are intuitively applied in the laboratory without strict formulation. Indeed, all relations of the laboratory must be explicitly stated in the model even if they are in reality only implicitly present

iv) Planning of urgent samples

Urgent samples are samples that obtain special treatment as compared to normal samples of the same type. They are analysed earlier, mostly because the results are urgently needed because of some emergency situation, e.g. to anticipate a halt in the production line. As to the laboratory organization, urgent samples have some important implications. First, the smaller delay times must be realized, which means that the activities of the technicians must be planned with a flexibility that allows some activities to be postponed until the urgent sample is processed. Second, the throughput of normal samples is affected, not only because of the postponement itself but also because of a decreased efficiency resulting from the unplanned activities of the urgent samples.

LABGEN can incorporate several mechanisms for urgent samples into a simulation model. The most elaborate mechanism occurs if centralized planning is combined with urgent samples which are to be analysed immediately. In this case a model structure is needed which is an intermediate between a continuous and a discontinuous model. The planner schedules the workload of normal samples for the whole day only in the morning. Urgent samples, however, are transmitted to the appropriate workstation immediately after their arrival in the laboratory. As a consequence, normal samples which are already scheduled, must make way and may be postponed until the next day.

v) *Handling of batches*

In many laboratories, samples can enter in batches, e.g. because several samples are taken at the same sampling point. These samples may be similar, requiring the same tests. They can also be complementary for investigating several aspects of the same composite object. Because this may affect the performance of the laboratory, LABGEN can incorporate both mechanisms of batch entrance into the simulation model.

A larger effect on the performance can arise from processing the samples in batches, a practice which is strongly stimulated by the application of autosamplers. This combination of samples allows several actions to be taken together, e.g. making of stock solutions, instrument tuning, calibration. This results in an overall reduction of processing time. To some extent, batch processing also has a buffering effect. If the workload is high, there are many samples in the laboratory that can be combined. The batches tend to become larger and, as a result, also the time savings. If the workload is small, the batches and consequently the time savings become smaller. Apart from statistical fluctuations, it is these buffering effects which make decision making in laboratories difficult.

Two major mechanisms for batch processing can be distinguished in laboratories, both of which can be modelled by LABGEN:

- Minimum batchsize: a batch of samples is only scheduled when its size exceeds a particular limiting value. The mechanism is useful to ensure that the capacity of the autosamplers, for instance, is optimally used. Larger minimum batchsizes result in larger time-savings, while on the other hand the delay times will increase.
- Sample combination: when a sample is scheduled, it is combined with other samples of the same type which happen to be present in the laboratory. The time-saving effect will be smaller but there will be no increase in delay times.

In both mechanisms the batchsize will have a maximum because of physical limitations of analytical equipment. Simulation experiments to demonstrate the effect of sample combination are discussed in [9].

vi) Samples consisting of several tests

Frequently, samples comprise more than one determination, which may be independent duplicate determinations or completely different tests. Because the analytical report for the customer depends on the completion of the last determination, this type of samples deserves special attention from the laboratory management. In order to study their effects on the laboratory performance, three different sample routing strategies are available in LABGEN:

- Sequential routing: the sample visits the workstation for the different tests one after another. After a test is completed, the sample moves to the next workstation. If not planned carefully, delay times of this type of samples tend to become rather large.
- Parallel routing: after entrance in the laboratory the sample is separated into aliquots for each test. The tests are scheduled independently and may even be performed at the same time.
- Parallel after preceding analyses: after one or more preceding analyses are completed, the other tests are routed in a parallel way. This is in fact an intermediate between sequential and parallel routing which is useful when for instance sample pre-treatment is necessary (this can be modelled as a separate test).

Parallel routing requires a careful administration, especially when many tests are performed on the same sample.

In order to demonstrate the effect of sample routing strategies, the models of Tables 2 and 3 are slightly altered: type1 (priority value 1 or due time 2 days) consists of two independent determinations. The total sample input stream is decreased in order to keep the workload of the laboratory at the same level. The results of the simulation experiments with these models are shown in Table 4. The delay times are larger if the two determinations are routed in a sequential way, especially in case of planning with priority values. Comparison with Tables 2 and 3 shows that the delay times are hardly affected in case of parallel routing.

Table 4

Effects of routing strategy on delay times. Same simulation models as in Tables 2 and 3 except that sample type1 consists of 2 independent determinations and that the input of samples is reduced by 1/3 (ratio type1/type2 = 0.5/0.5). Between parentheses: standard deviation of the distribution.

		delay times (in days)		
		all samples	type1	type2
prio.	seq.	1.84 (2.54)	3.79 (2.43)	0.00 (0.00)
	par.	1.04 (1.66)	2.12 (1.83)	0.00 (0.00)
due	seq.	1.81 (1.61)	2.80 (1.47)	0.83 (1.03)
	par.	1.19 (1.31)	1.78 (1.38)	0.61 (0.92)

vii) Interaction between technician and instrument

The analytical process in the laboratory can be considered as an interaction between technician and instrument, with the sample as an object. Clearly, the character of this interaction is essentially important for the performance of the laboratory, because it forms its very basis. The interaction can be addressed at two levels: at the level of instrument operation and at the level of sample processing.

Two extreme modes of technician-instrument relations are:

- Each technician is confined to only one instrument without regard to the workload of other instruments. This has the advantage that the analysis jobs are carried out with a maximal efficiency. However, the laboratory becomes very inflexible and easily affected by "disorders" such as absence of technicians because of illness etc.

- Each technician can operate all instruments and the choice of an instrument depends on the workload. At the expense of efficiency this strategy will give rise to an increased flexibility. A situation where one part of the laboratory cannot handle the workload while another part is free of work, is thus avoided. Because various technicians analyse similar samples, it becomes more difficult to combine activities.

In most laboratories an intermediate between these extremes is employed, e.g. a technician may operate only a group of similar instruments. This is a very important factor which cannot be decided on from laboratory performance considerations only. Also experience and qualification of the technicians must be taken into account. Both extreme and intermediate types of technician-instrument relation can be modelled by LABGEN.

viii) Sample processing

At the level of a specific test being carried out with an instrument, several steps can be distinguished. Although these steps are largely determined by the analytical method, some aspects can be subjected to an optimization to organizational criteria. The following five steps are used by LABGEN to model sample processing:

- 1 Sample preparation: several chemical actions for which the instrument is not yet needed, e.g. dissolving or diluting the sample, sub-sampling and so on;
- 2 Instrument tune-up: preparation of the instrument for the analysis and calibration;
- 3 Sample measurement: the actual measurement of the sample. In many instances, the technician is free to perform another job during the measurement time;
- 4 Closing: e.g. cleaning of the instrument after the measurement. This activity is only dependent on the instrument;

5 Sample report: calculation of the analytical result and writing a report for the customer. For this activity the instrument is not needed anymore.

Step 2 and 4 are optional for each instrument. In fact they are not included in the previously described basic model.

Optimization of the laboratory performance must include a careful consideration of these sample processing steps. Batch processing of samples has a large effect as has been discussed above. For reasons of efficiency, it may be useful to collect a number of report jobs before the reporting is actually carried out or to restrict the reporting to a particular time or weekday. It will be clear that delay times are directly affected by such decisions. However, the effect on the total laboratory performance is less clear. Several alternative modes can be modelled by LABGEN.

ix) Stand-by technicians

Most laboratory managers will at least sometimes have the feeling that their laboratory is permanently understaffed. In most laboratories it is simply too expensive to employ more than a minimal number of technicians. The consequence of this is that situations will occur that the laboratory cannot master the sample stream and needs a higher working capacity. This capacity can be increased by overwork of the technicians or by a temporary increase in number of technicians. They may be called in from another department or from a temporary employment agency.

These mechanisms greatly add to the flexibility of the laboratory. It is very important to simulate the laboratory in the way it responds to changing circumstances. Therefore, LABGEN is able to model both overwork and stand-by technicians. However, it is difficult to formulate the criteria needed to decide to temporarily employ an additional technician. This is necessary in order to apply this simulation feature in a model. In a real laboratory these decisions are usually made on an intuitive basis. For incorporation into a specific model it is expected that a

first guess of the relevant parameters has to be made. Their proper settings then are to be realized by means of a tuning process.

x) Disturbances

Decision making in laboratories is complicated by statistical fluctuations in sample input and processing times, buffering effects caused by batch processing for instance, and disturbances. Disturbances may be caused by instrument breakdown or by absence of technicians because of illness. They have a statistical nature. Their occurrence sets constraints to the flexibility of the laboratory because each disturbance must be met in order to realize a satisfactory performance.

The way a laboratory reacts to disturbances depends on its specific organization. For instance, if an instrument can be operated by only one technician, problems will arise when this technician is absent and no one can take over. Therefore, it is very important for each possible decision concerning the laboratory organization to take into account the effect of possible disturbances.

In order to make this possible, LABGEN can incorporate disturbances into the simulation model, both for technicians and instruments. This means that each object of the lab component in the model obtains its own disturbance process. This is determined by a mean inter-disturbance time and a mean disturbance time. During a disturbance the instrument or technician in fact does not belong to the system, i.e. it cannot perform any activity and no activity can be planned for it.

The possibility of machine breakdown is incorporated into the models of Tables 2 and 3 (ratio of input type1/type2 = 0.5/0.5). The frequency of the disturbances for both instruments is once in 40 days with an average disturbance time of 2 days. This means that each instrument is not available during about 5 % of the time. The results of the simulation experiments are listed in Table 5 and Figure 7. Although in the models with disturbance the input was decreased by 5 %, the mean delay times are increased significantly, especially type1 in case of plan-

Table 5

Effects of instrument breakdown on delay times. Same simulation models as in Tables 2 and 3 except that an instrument is not available during about 5 % of the time and that the input of samples is reduced by about 5 % (ratio type1/type2 = 0.5/0.5). Between parentheses: standard deviation of the distribution.

		delay times (in days)		
		all samples	type1	type2
prio.	normal	1.11 (1.84)	2.32 (2.10)	0.02 (0.15)
	disturb.	2.21 (4.72)	4.71 (6.54)	0.29 (0.63)
due	normal	1.10 (1.14)	1.65 (1.19)	0.59 (0.80)
	disturb.	2.23 (2.73)	3.09 (3.03)	1.44 (2.12)

ning with priority values. The effects of the disturbances are almost only accounted for by the low priority samples. If this additional spread in delay times is not acceptable in a laboratory, stand-by instruments are necessary.

DISCUSSION

Only the most important aspects of the laboratory organization are discussed above. Several other options can be chosen during the construction of a simulation model, such as the types of distribution of the specific random procedures in the model.

The laboratory organization is determined by a large number of parameters. Many qualitative (descriptive) and quantitative data about

Figure 7. Distributions of the delay times of sample type of the model of Table 2 with ratio type1/type2 = 0.5/0.5. a) no instrument breakdown; b) instrument breakdown: mean inter-breakdown time is 40 days, mean duration of each breakdown is 2 days.

the laboratory are required for the construction of a simulation model. This observation leads to an interesting application of LABGEN:

- 1) Application as a checklist: information requested by LABGEN shows which parameters are important to be considered for the specific kind of laboratory.

At this level, simulation experiments are not yet performed. Three other levels of application can be distinguished:

- 2) Application for more theoretical experiments: simulation experiments are performed to study the effects of variations to a parameter in a more general context. The experiments discussed above can serve as examples of application at this level.
- 3) Application to parts of a specific laboratory: a full simulation cycle is performed, consisting of model construction, validation and simulation experiments, but only one or a few workstations are considered. This application is useful if only a part of the laboratory needs to be considered which is relatively isolated from the rest.
- 4) Application to a laboratory as a whole: a full simulation cycle concerning the whole laboratory organization is performed. In Part II of the present series of papers [15] an application at this level is discussed.

The current version of LABGEN is implemented on a mainframe computer (NAS 9600). Two future developments are very interesting. First, incorporation of LABGEN into a LIMS environment, which would transform the LIMS from an information management system into a laboratory management support system [2]. Second, implementation on a personal computer would greatly add to the portability of LABGEN, and thus to the applicability. Current PC's using MS-DOS as an operating system are not yet suitable for LABGEN (although a SIMULA compiler for MS-DOS is available). However, the OS/2 operating system for PS/2 personal computers likely will provide an environment which is suitable for LABGEN.

REFERENCES

- 1 D L Massart, B G M Vandeginste, S N Deming, Y Michotte and L Kaufman, *Chemometrics a Textbook*, Elsevier, Amsterdam, 1988
- 2 J Klaessens, B Vandeginste, G Kateman, Towards computer supported management of analytical laboratories, *Anal Chim Acta*, in press
- 3 J A Payne, *Introduction to Simulation Programming Techniques and Methods of Analysis*, McGraw-Hill, New York, 1982
- 4 A M Law and W D Kelton, *Simulation modeling and Analysis* McGraw-Hill, New York, 1982
- 5 P Winkel, *Clin Chem*, 30 (1984) 1758
- 6 T A H M Janse, G Kateman, *Anal Chim Acta*, 159 (1984) 181
- 7 B van de Wijdeven, J Lakeman, J Klaessens, B Vandeginste, G Kateman, *Anal Chim Acta*, 184 (1986) 151
- 8 J Klaessens, J van Schalkwijk, P Cox, R Bezemer, B Vandeginste, G Kateman, *J Chemometrics*, 3 (1988) 81
- 9 J Klaessens, T Saris, B Vandeginste, G Kateman, *J Chemometrics*, 2 (1988) 49
- 10 M S Elzas, Relations between artificial intelligence environments and modelling & simulation support systems, in M S Elzas, T I Oeren, B P Zeigler (eds), *Modelling and simulation methodology in the artificial intelligence era*, Elsevier, Amsterdam, 1986, p 61
- 11 R O'Keefe, *Simulation*, 46 (1986) 10
- 12 F J Wales, P A Luker, Design of a multi-level user interface for discrete event simulation, in G C Vansteenkiste, E J H Kerckhoffs, L Dekker, J C Zuiderv-aart (eds), *Proceedings of the 2nd European Simulation Congress, SCS, San Diego*, 1986, p 361
- 13 J Haddock, *Simulation*, 48 (1987) 45
- 14 G M Birtwistle, O J Dahl, B Myhrhaug and K Nygaard, *SIMULA Begin*, Petrocelli/Charter New York, 1973
- 15 J Klaessens, J Sanders, B Vandeginste, G Kateman, LABGEN, expert system for knowledge based modelling of analytical laboratories Part II Application to a laboratory for quality control, *Anal Chim Acta*, submitted for publication
- 16 B G M Vandeginste *Anal Chim Acta*, 112 (1979) 253
- 17 B G M Vandeginste *Anal Chim Acta*, 122 (1980) 435

Part 4

APPLICATION OF SOFTWARE MODULES TO A LABORATORY FOR QUALITY CONTROL

SUMMARY

A simulation project is described to evaluate the performance of a section of a laboratory for quality control. The simulations are carried out with two recently developed software modules. The simulation model is automatically constructed by the expert system LABGEN. The preliminary processing of historical laboratory data stored in the LIMS is carried out with the programs of the module FEA. The results of the simulation project are discussed.

INTRODUCTION

Digital simulation is a valuable method for decision support in analytical laboratories. In many cases application of the method was too expensive and time consuming to be of practical value. In order to overcome these drawbacks, the software modules FEA and LABGEN have been developed. A detailed description of the programs is published elsewhere [1-3]. In Part I of the present series of papers the laboratory organization aspects of LABGEN are discussed [4]. This paper, Part II, discusses the application to a laboratory for quality control.

The application not only serves to provide decision support in a specific laboratory situation. It also serves as a testcase for the software modules. Are they flexible enough to deal with the complexities of a real laboratory? Can a model constructed by LABGEN be used to evalu-

* This chapter is published as: J. Klaessens, J. Sanders, B. Vandeginste, G. Kateman, LABGEN, expert system for knowledge based modelling of analytical laboratories. Part II: Application to a laboratory for quality control, *Anal. Chim. Acta*, submitted for publication

ate possible decisions? The present paper describes the laboratory situation and the results of the simulation project. Also, the application of LABGEN is evaluated.

THE LABORATORY SITUATION

The laboratory considered is the section for inorganic analysis of a laboratory for water quality monitoring which is situated near a municipal potable-water plant. This laboratory carries out analyses for this plant and for many external customers, e.g. other water treatment works, sewage purification plants, swimming pools and projects for investigation of polluted areas. Especially the samples from external customers cause planning problems because of the fluctuations in supply which can hardly be foreseen.

In the section approximately 70 tests are more or less regularly carried out on water and soil samples. Depending on the workload the section is staffed with 3-4 technicians. Initially - the 1987 situation - the measurements were conducted on two AAS instruments (flame and furnace). Since then, however, an important enlargement of the equipment has been realized by the introduction of an AAS instrument with Zeeman detection and an ICP instrument.

Another important development in the section is the increase in samples from external customers and a corresponding increase in urgent samples. Urgent samples are samples whose analytical results are requested within one or two days. In the laboratory the tests are carried out in batches. As has been observed before [5,6], in this kind of laboratory the delay times of the samples are strongly dependent on the applied batchsizes. The batchsize determines the actions that can be combined (making of stock solutions etc.), and thus the time savings. Immediate processing of batches with urgent samples tends to decrease the overall batchsize and thus has an important effect on the performance of the laboratory.

The timely deliverance of the analytical results of these urgent samples is considered to be an important service for the customers. In order to shed some more light on the impact of abovementioned developments on this service a simulation project has been carried out.

THE SIMULATION PROJECT

LABGEN and FEA offer the possibility to quickly develop a simulation model. This allows the analyst to monitor the performance of the laboratory section. In order to do so, simulation models for three different periods are constructed.

- 1) Model 1 (initial situation) based on the historical datafile from 19 March until 30 June 1987, 2 AAS instruments, 3-4 technicians
- 2) Model 2 (after introduction of AAS with Zeeman detection): based on laboratory data from 6 January until 31 March 1988, 3 AAS instruments, 3 technicians
- 3) Model 3 (after introduction of ICP instrument): based on laboratory data from 20 June until 9 August 1988; 3 AAS and 1 ICP instruments.

The construction of a model consists of a session with FEA to process the laboratory data and a session with LABGEN to build the simulation model. Data processing consists of data reduction (modelling into a limited number of sample types) and of a signal quantification (e.g. input in samples per day, type distribution). The quantification step is carried out independently for each model. The data reduction step, on the other hand, is carried out for the first model only. For the subsequent models it is checked whether the results can still be applied. Also the first simulation model of the laboratory is used as a starting point for the other two models. The modelling process is described using model 1. For the other models the necessary adaptations are described. The validation process and the simulation experiments, however, are mainly described for model 3.

FEA, LABGEN and the automatically constructed simulation models are written in SIMULA [7]. They run under the VM-CMS operating system of a NAS 9600 mainframe computer.

CONSTRUCTION OF THE MODEL

Model 1

The data processing with FEA for model 1 has been discussed in detail in [2]. The results need only be outlined here. By combination the number of different tests was reduced from 69 as present in the datafile to 21. Apart from these tests, 2 different sample pretreatment methods are used. The datafile contained 122 different patterns of pretreatment and tests, 56 of which occurred only once. These patterns could be classified in 3 sample types:

- 1) type 1: samples without pretreatment (water);
- 2) type 2: samples with pretreatment code 13 (soil);
- 3) type 3: samples with pretreatment codes 13 and 14 (soil).

The sample types contain a set of different tests, each having a specific chance of occurrence. Three distinct sample origins could be discerned. Samples from different origins obtain different treatment even if they are of the same type. The laboratory input depends on the weekday (Table 1). Besides this week effect no correlation was present in the input signal.

Many necessary quantitative data could not be obtained from the datafile. These are estimated in sessions with the laboratory staff. The most important are:

- The processing times of the different tests. These times are largely determined by the instrument used. Sample pretreatment is carried out on 12 refluxing units. They are operated in sets of 6 units. Therefore, they are modelled as 2 instruments.

Table 1

Real sample input of the total laboratory specified to the weekday. The values are from the period used for the construction of model 1

model 1	

Monday	25.7
Tuesday	36.2
Wednesday	40.0
Thursday	32.9
Friday	24.6

Mean	31.8

- The fraction of urgent samples. This is estimated as: 10 % of types 2 and 3; 1 % of type 1.
- The working times of the technicians. Only 2 of the 4 technicians are full time available.

The values thus obtained may be inaccurate and may require further refinement in order to arrive at a proper model. This refinement is part of the validation procedure.

The simulation model is actually constructed in an interactive run of LABGEN. Several refinement steps are performed before a close resemblance between model and reality was realised. The final settings for model 1 are included in the Appendix. It shows the dialogue that would lead from the start to the final model 1. For reasons of length the quantitative data are not included. This dialogue did not actually take place, because the first dialogue contained assumptions that proved wrong and are changed using LABGEN's back-up facilities. The major features of the laboratory constructed this way are outlined below.

Most samples require several tests to be performed (elements to be determined). They may enter the laboratory in batches of samples with the same tests. Samples of types 2 and 3 must first be pretreated before the tests can be carried out. After pretreatment (if necessary) the tests of the sample are divided among the different batches awaiting analysis. A batch is only analysed if at least the minimum batchsize is present. An exception is made for urgent samples. These are analysed as quickly as possible. They are of course combined with the other tests of the same kind that are waiting.

The laboratory does not have central planning. This means that the work to be done during the day is not centrally planned by the laboratory head in the morning. It is the technicians who choose which batch of tests is processed next. This depends on the batchsize, urgency, and due time of the tests in the batch. It has already been reported that in laboratories with batch processing the delay times of the samples are highly dependent on the minimum batchsizes employed [5,6].

Only two technicians can operate all instruments; the other two technicians are assigned to sample pretreatment only. The instruments have different modes of operation. The furnace AAS makes use of an auto-sampler, allowing the technician to do other jobs during the measurement time of a batch. This is also the case for sample pretreatment during the refluxing time. As mentioned before, this is modelled as an instrument. The flame AAS, on the other hand, must be attended to by the technician during the measurements, because additional dilutions may be necessary.

The validation of the model consists of different steps. First, there may be no statistically significant difference between the real and simulated sample input. Second, the model settings such as the processing times of the instruments must be approved of by the laboratory staff. Third, there may be no statistically significant difference between the real and simulated delay times of the samples. For model 1 steps 1 and 2 are successfully carried out, although a detailed description is only given for model 3. Step 3 is carried out only globally, because the real delay times of the inorganic section were not available for this period. Only the delay times for the total laboratory were available including

authorisation which is done only once a week. The real delay times were available only for model 3.

Some simulation experiments are carried out with model 1 although important changes had already taken place in the laboratory section. These experiments, however, clearly showed that the situation was rather unstable. A slight increase in the input caused a large increase in delay times. The laboratory could hardly face the workload. This result was recognized by the staff, which can be interpreted as an indication of the validity of the simulation model.

Model 2

The second model of the laboratory is based on the datafile from the period 6 January until 31 March 1988. For this model several changes had to be incorporated. A new furnace AAS with Zeeman detection was introduced. This instrument has the same processing characteristics as the first furnace AAS. There were only 3 technicians available, instead of 4 in model 1. An important new activity introduced in the laboratory model is grinding of the soil samples. In the previous model, grinding was carried out outside the laboratory section considered. All soil samples must be ground before they can be pretreated. This activity is modelled as two different tests: short grinding which takes approximately 10-15 minutes and long grinding which takes about 1 hour. The latter activity comprises about 30 % of the soil samples. These are actually composed of 5-6 combined soil samples. This results in a considerable decrease in sample input, as is shown in Table 2 in comparison with Table 1.

Another activity which is added to model 2 is filtering. About 15 % of the water samples must be filtered before the elements can be determined. This activity requires approximately 15 minutes and can be modelled as a test preceding all other tests on the sample. It had erroneously been neglected in model 1, indicating that the situation in this period was even worse than observed in the simulation experiments. Figure 1 shows a general view of the sample flow through the laboratory section.

Table 2

Real sample input of the total laboratory specified to the weekday The values are from the periods used for the construction of models 2 and 3

	model 2	model 3
Monday	11.5	13.5
Tuesday	28.0	37.9
Wednesday	22.8	23.1
Thursday	17.4	11.1
Friday	18.1	14.4
Mean	19.6	20.0

Figure 1 Sample flow through the section for inorganic analysis

The input model of model 1 was also valid for model 2. 3 sample generators, 3 distinct sample types All numerical aspects are adapted to the new situation as analysed with the FEA programs. Model 2 is vali-

dated in the same way as model 1, which implies that the first two steps of the validation procedure are carried out successfully and that the third step with the delay times is carried out globally. Model 2 has been used for a number of simulation experiments. They are discussed below along with the experiments with model 3.

Model 3

The third model is based on data from the period 20 June until 9 August 1988. The development of model 3 from model 2 required fewer changes than the development of model 2 from model 1. A new ICP instrument was introduced. This instrument is able to analyse several elements at the same time (4-6), that were formerly analysed on the flame AAS (sample type 2). The code 13 pretreatment is no longer carried out by 2.5 hour refluxing. A microwave oven is used instead, requiring only 1.5 hour. The grinding is carried out in batches of 4. Apart from these changes also the numerical aspects of the sample input and the working times of the technicians had to be adapted to the new situation.

For this period the real delay times of the inorganic section were available, independently from the other sections visited by the sample and the authorisation carried out only weekly. This allows a complete model validation, containing as a third step the test whether the real and simulated delay times belong to the same distribution. Because of this and also because of the fact most simulation experiments are carried out with model 3, it is justified to discuss the validation in more detail.

For the validation 9 independent simulation runs are carried out with the model. It is tested whether the values of the real laboratory may belong to the distributions obtained from these runs. Figure 2 contains the mean values with the 95 % confidence interval of the individual measurements, based on the Student t distribution ($\alpha=0.05$, 8 degrees of freedom) as calculated from the simulation runs. It is important to realize that the inputs for the generators are independent variables. The

Figure 2. Comparison of the real values of the laboratory input (in samples/day) and the 95 % confidence interval calculated from 9 independent simulation runs; + (bold): mean value of the runs; + (normal): bounds of 95 % confidence interval; o real value. a) specified to sample generator, b) specified to weekday

same applies for the inputs as specified to the weekdays. Except for the Fridays, the real values lie within the confidence intervals. The Fridays constitute only a minor part of the input. With 95 % confidence intervals it must always be expected that 5 % of the values exceeds the interval bounds. Therefore, it is justified to reject the alternative hypothesis H_1 that there is a significant difference between real and simulated input. The settings of the model parameters, such as the processing times of the instruments, are approved of by the laboratory staff. This is an important part of the validation process.

As for the third step - the testing of the delay times - it must be realized that the delay times of the various sample types are not indepen-

Table 3

Comparison of real and simulated (mean values of 9 runs) delay times as specified to sample type. The values in parentheses represent the standard deviation calculated from the 9 runs. ¹: estimated by the laboratory staff.

	simulated	real
Total	5.27 (0.36)	5.4
Type 1	5.19 (0.34)	5.1
Type 2	4.73 (0.69)	4.7
Type 3	6.58 (0.53)	7.4
Urgent samples	1.63 (0.20)	1-2 ¹

Figure 3. Distribution of delay times for model 3. a) real laboratory, b) mean of 9 simulation runs.

dent. Therefore, only the overall mean delay time is subjected to the Student t test. From the 9 simulation runs the 95 % confidence interval

has been calculated for this variable. [4.44 - 6.10]. The real value is 5.4, which is well within the confidence interval. In Table 3 the real and simulated delay times of the different sample types are compared. Figure 3 shows both distributions of delay times. The fact that model 3 could be developed from model 2 by the introduction of the ICP and the microwave oven but without any changes in the other model settings being necessary (except of course the laboratory input characteristics) confirms that the model is a good reflection of the reality.

APPLICATION OF THE MODELS

Most simulation experiments are carried out with model 2 and especially model 3. For the experiments parameter settings are chosen that are not too far off the actual values. The more deviation from reality is introduced the less confidence can be placed in the results.

Table 4

Factors and the levels of the 4^2 factorial design with model 2

		+ level	- level
A	min. batchsize	type 1: +33 %	type 1: -33 %
	of instruments	type 2/3: +20 %	type 2/3: -20 %
B	lab. input	+20 %	-20 %
C	work. capacity	+1 techn. day	-1 techn. day
D	urgent samp.	normal	-45 %

Table 5

Results of the 4^2 factorial design with model 2. n is the number of duplicates at each experimental setting ($n = 2$). Only significant effects are shown: ¹: significant with $\alpha = 0.05$; ²: significant with $\alpha = 0.01$. Factors are explained in Table 4. Standard deviation (s) is calculated from 10 independent runs with model 2.

	A	B	C	D	$s/\sqrt{n}$
Total delay	0.86 ²		-0.54 ¹		0.17
delay type 1	0.96 ²				0.20
delay type 2	0.75 ¹	0.60 ¹	-0.74 ¹		0.25
delay type 3		0.97 ¹	-0.78 ¹		0.34
delay urgent	1.04 ²		-0.50 ¹		0.21
busy times		0.81 ²	-0.59 ¹		0.20

A technique frequently applied in simulation to guide experimentation is factorial design (experimental design) [8,9]. With model 2 two 4^2 factorial designs (4 factors, 2 levels) were used. The first design was used during the validation phase to check for instance the effect of the choice of the type of random distribution of the input generators. The second design was used to study the effects of some important factors on the performance of the laboratory. The levels chosen - a slight increase resp. decrease of the initial value - are listed in Table 4. At each experimental setting duplicate experiments are carried out. The results are shown in Table 5; only the significant effects are listed. None of the interaction effects appeared to be significant. As a matter of fact, this is the most important conclusion drawn from these experiments, as it allows us to study the effects of the factors independently. However, it is reassuring to find a large effect of the minimum batchsize. A batch of samples can only be processed if its size exceeds a certain minimum

value. In other simulation experiments [5,6] it was observed that this factor is very important for the delay times. Another striking result is that the fraction of urgent samples does not have any significant effect. This is one of the factors that is studied in more detail with model 3.

Increased sample input

One of the factors in the factorial design with model 2 was the laboratory sample input. The effect of this factor is studied with model 3 at a number of levels of increased input. The highest level was a mean increase of 2.9 samples/day (initial value: 14.2). At each level 4 independent simulation runs are carried out. The results are shown in Figure 4, where for each level the 95 % confidence interval of the mean is depicted. The results fit very well with the results obtained with the factorial design with model 2.

The delay times of sample type 1 are not affected by the increased input; the delay times of urgent samples are hardly effected. Types 2 and 3 show distinct effects. For these types the dispersion tends to become very large at large input levels. The performance becomes unstable: at these levels the laboratory is no longer able to handle the workload in any case. Likely, sample pretreatment imposes too large a workload on the technicians. As for type 1 which does not require pretreatment, the additional demand can easily be met by combination into the batches. This results in an increased mean batchsize, which was actually observed in the simulation experiments. The urgent samples are not affected by the increased input although most of these samples require pretreatment. However, they always have priority before normal samples. The laboratory is flexible enough to give them this special treatment.

Figure 4 Delay times as a function of additional laboratory sample input. The initial input is 14.2 samples/day. The 95 % confidence intervals of the mean values of 4 runs are depicted. a) sample type 1; b) sample type 2; c) sample type 3; d) urgent samples

Work capacity

The unstable performance at high input levels is not caused by lack of equipment. At the highest input level of Figure 4 several experiments are carried out with additional work capacity. Starting at the initial level of 10 technician days per week (from 3 technicians), the work capacity is increased with steps of 0.5 day until 12 technician days per week. As

is shown in Figure 5 the delay times (the overall value as well as the value of urgent samples) diminish again. As a matter of fact, this result is not surprising because of the increase of equipment during the preceding year.

Figure 5. Delay times as a function of additional work capacity (initial: 10 technician days/week). The laboratory input is at the highest value of Figure 4, 17.1 samples/day. The 95 % confidence intervals of the mean values of 4 runs are depicted. a) all samples; b) urgent samples.

With additional manpower the laboratory is able to handle an increased sample input. What is the situation if the same workload must be handled with reduced manpower? Figure 6 shows the results of experiments with reductions in manpower up to 15 %. The delay times become larger although the increase of the means of the 4 independent runs is not dramatic. The effect is rather that the performance becomes more unstable. It can no longer be guaranteed that the laboratory can handle the workload. At the same time also an increase in busy times of the technicians of approximately 15 % has been observed. The conclusion, therefore, must be that any reduction in work capacity has an

Figure 6 Delay times as a function of reduction of work capacity (initial 10 technician days/week) The laboratory input is at the normal level 14.2 samples/day. The 95 % confidence intervals of the mean values of 4 runs are depicted a) all samples, b) urgent samples

immediate negative effect on the performance of the laboratory, although not of a dramatic size

Urgent samples

The factorial design with model 2 showed that the present fraction of urgent samples has no significant effect on the delay times. The impact of an increase of urgent samples up to 220 % is displayed in Figure 7. The total laboratory input is kept constant, only the fraction of urgent samples is increased. In the highest level 22 % of types 2 and 3 are urgent and 2.2 % of type 1. The striking result of these experiments is that there is hardly any effect. Especially type 1 and the urgent samples show no effect. The performance with respect to types 2 and 3 may have become somewhat unstable. However, comparison with the profiles of Figure 4 for instance indicates that this is not a clear effect. Figures 7 b

and c contain the delay times of all type 2 and type 3 samples, including the urgent samples. Constant delay times in the figures imply that the delay times of non-urgent samples become somewhat larger with an increasing fraction of urgent samples. These small increases are actually observed.

Figure 7 Delay times as a function of increased number of urgent samples. The initial value is 0.67 urgent samples/day. The total input is not affected. The 95 % confidence intervals of the mean values of 4 runs are depicted. a) sample type 1, b) sample type 2; c) sample type 3, d) urgent samples.

It was expected that the increase of urgent samples would cause a reduction in the overall batchsize. This would have a negative effect on the performance because the possibility to combine activities would be reduced. However, only a very small effect was observed. Also the increase in working times of the technicians was very small. Probably the operation was determined by the presence of urgent samples already at the initial level. With a batchsize of 15 (which is the actual mean value) and a fraction of urgent samples of 10 %, about 80 % of the batches contain at least one urgent sample. Indeed, experiments without urgent samples showed a larger increase in batchsize than the decrease observed in abovementioned experiments. Also an increase in overall delay time was observed.

The laboratory is very flexible with respect to urgent samples. An increase of 220 % can be handled without difficulty.

CONCLUSION OF THE SIMULATION PROJECT

The validation showed that the simulation models are a good reflection of the reality. The simulation experiments indicate that in the present situation the laboratory is very well suited for its task. It has enough flexibility to respond to changes in the sample input. Relatively large fluctuations in the input can be easily handled. If a high input continues for a longer time, the workload may become too large. However, the capacity of the equipment is such that with additional manpower a satisfactory performance can be realized.

The flexibility of the laboratory is also clearly demonstrated by the way it responds to an increased number of urgent samples. Usually, these have a de-stabilizing effect because they require a deviation of the normal processing order. The already planned activities must be postponed until the urgent sample is processed and the possibility to combine activities is reduced. However, at least at the levels investigated these de-stabilizing effects have not been observed. A relatively large increase of urgent samples can be handled, almost without negative

effects on the delay times of the non-urgent samples and the working times of the technicians. This is an important conclusion with respect to the services offered to the customers.

The three subsequent simulation models show a development from a laboratory hardly able to face the workload to a laboratory that can handle it with great flexibility. At the same time the necessary manpower is reduced. This is accomplished not only by the introduction of additional equipment, but also by new procedures. The grinding procedure allows several soil samples to be combined. This wrongly gives the impression of a reduced sample input (Tables 1 and 2). Application of the microwave oven made possible a reduction of pretreatment time of soil samples.

Although usually not to the same extent as the laboratory under consideration, laboratories are constantly changing. As a consequence, simulation of a long period (e.g. 1 year) is in many cases too long. Simulation models covering shorter periods (several months) offer better opportunities for decision support if the simulation project can be carried out in a reasonably short time. The results must be available before changes have already taken place. LABGEN and FEA enable the analyst to quickly develop a new model, especially if a former model can be used as a starting point. The development of model 3 and the subsequent experimentation required only a few days. Another advantage is the gradual improvement of the model.

It has been suggested [8] that once a simulation model has been constructed, it can be easily kept up to date by rewriting the relevant parts of the computer code constituting the model. The developments observed in the laboratory under investigation clearly demonstrate that this is not correct. So many parts of the code would have to be adapted that the construction of a completely new program likely would be a simpler task. With LABGEN these changes are easily introduced. LABGEN always constructs a new simulation program with the - possibly changed - data concerning the laboratory. All relations and links between laboratory parts are in accordance with these data.

EVALUATION OF LABGEN

The simulation project is not carried out by the laboratory staff itself because the applicability of LABGEN had to be tested first. Because the program requires much interaction with the user, it is important to investigate how well it can be used by the less-experienced user. Are the questions posed clear and are the help facilities offered sufficient? Can the user foresee the implications of his answer? Likely, some adaptations concerning these aspects of the modules will appear to be necessary.

The application of LABGEN to the laboratory section for inorganic analysis was preceded by an extensive debugging procedure in order to remove model inconsistencies causing run-time errors during the simulation run or causing model activities to run in a different way as intended. During the simulation project this testing continued. Currently, LABGEN is rather stable, i.e. errors are no longer detected.

In the simulation project LABGEN and FEA showed to be flexible enough to construct a validated simulation model of a real laboratory and to carry out a performance evaluation. No prior knowledge about the specific laboratory was available during the development of LABGEN. During the simulation experiments some adjustments to LABGEN appeared to be useful, mostly minor generalizations to allow a more detailed specification of the sample input stream. The most comprehensive adjustment was the introduction of the concept sample pretreatment: a number of tests belonging to the same sample can be processed in a parallel way only after a preceding pretreatment is completed. All introduced adjustments have significance in general. It is expected that only few adjustments will appear to be useful during future applications to different types of laboratories.

Acknowledgement

The authors thank Mr. E. van Naerssen and Mr. W. Gademan of N.V. Regionaal Energiebedrijf Dordrecht (The Netherlands) for providing the laboratory data and for their cooperation in the research.

APPENDIX: MODEL SETTINGS OF MODEL 1.

The answers given by the user to develop model 1. They are ordered according to lab component. The quantitative data are not included for reasons of length.

Main

Enter the number of sample types. → 3

Are there any sample types with more than 1 analysis requested? → Y

Enter the number of specific analyses (determinations).
→ 23

The laboratory is able to process urgent samples? → Y

Are the delay times of urgent samples larger than 1 day?
→ Y

The week-ends are discarded for time calculation during the simulation run? → Y

Histograms of performance specified according to urgent / not-urgent? → Y

Histograms of delay times specified according to sample type? → Y

Histograms of delay times specified according to generator? → N

Histograms of performance specified according to generator? → N

Do you want to initialize the main level ? → Y

Generator

Enter the number of sample generators. → 3

The input of the generators is dependent on the weekday?
→ Y

Enter the type of distribution(s) of the number of sample entries per day.

- a) poisson dist.
- b) uniform-integer dist.
- c) histogram shaped dist.
- d) fixed values. → C

Samples can enter the laboratory in batches? → Y

Are there any sample types that only enter the laboratory together with one or more other types? → N

Do all analysis types have the same distribution of preparation times? → N

Choose distribution for preparation times:

- a) normal dist.
- b) uniform dist.
- c) negative exponential dist. → B

Do all analysis types have the same distribution of measurement times? → N

Choose distribution for measurement times:

- a) normal dist.
- b) uniform dist.
- c) negative exponential dist. → B

Do all analysis types have the same distribution of report times? → N

Choose distribution for report times:

- a) normal dist.
- b) uniform dist.
- c) negative exponential dist. → B

How should the separate analyses of sample with more than 1 analysis be routed?

- a) in a sequential way;
- b) in a parallel way (i.e. independent);
- c) parallel after one or more preceding analysis. → C

How are the analyses of a sample determined?

- a) each type occurs only once and in fixed order
- b) each type occurs only once in random order
- c) random; each type may occur more times; same chance.
- d) as c; types have different chances. → D

The number of analyses per sample is type dependent. It will be determined:

- b) random according uniform-integer distribution
- d) fixed number. → B

The distributions concerning batch entrance are equal for each sample type? → N

Choose the type of distribution for batch size of entries:

- a) poisson dist.
- b) uniform-integer dist.
- c) histogram shaped dist.
- d) fixed values → C

Do you want to initialize the generator ? → Y

Planner

The planning of sample scheduling is dependent of the sample type? → Y

The priority of a sample is determined by its planned due time only? → Y

Distribution of planned due time for each entering sample:

- a) poisson dist.
- b) uniform-integer dist.
- d) fixed values. → D

Scheduling of urgent samples differs from normal ones? → Y

Urgent samples are planned with:

- a) fixed values;
- b) introduction of an urgent factor. → A

Urgent samples have always priority before non-urgent samples? → Y

Urgent samples should immediately be scheduled, even if the minimum batchsize is not yet exceeded? → Y

Is only one instrument suitable for a particular analysis? If 'n' then several instruments may be suitable. → N

The actual analysis jobs are planned by the planner? → N

The planner assigns samples to a technician? If 'no' then assignment to an instrument. → N

The workload of the workstations depends on:

- a) number of samples;
- b) total needed analysis time;
- c) total needed analysis and preparation time. → A

A sample should be assigned to the allowed workstation which has the smallest workload? If no: it is assigned to the first of the list of allowed workstations. → Y

Analyses can be analysed in batches? → Y

A batch is only scheduled if a minimum size is exceeded? → Y

Reduction factors for preparation, measurement and reporting are dependent on the sample type? → Y

Do you want to initialize the planner ? → Y

Technician

Enter the number of technicians (if relevant: including reserve technicians). → 4

A technician can operate all instruments? → N

The model has one or more reserve technicians which are employed in times with a high workload? → N

Reporting can be done:

- a) only first job in the morning;
- b) reporting at any time of the day. → **B**

Reporting jobs are gathered? If no. reporting immediately after analysis → **N**

Technician may be struck by sudden illness in the model?
→ **N**

A technician must finish an already started analysis job at the end of the day? → **Y**

A technician should take the (allowed) instrument having the highest priority analysis job? If no: take the first (allowed) instrument available. → **Y**

Apart from a histogram for total worktimes of each technician, also histograms for measurement and reporting times per day? → **N**

Do you want to initialize the technicians? → **Y**

Instrument

Enter the number of instruments. → **5**

Instrument breakdown is allowed in the model? → **N**

Should the adjust time be taken into account for any of the instruments? → **Y**

Should the closing time be taken into account for any of the instruments? → **N**

The adjust time is a fixed value ? → **N**

The distribution of adjust time is:

- a) normal;
- b) uniform,
- c) negative exponential. → **B**

Which instrument(s) have the adjust property ? → **A B C D**

Do you want to initialize the instruments ? → **Y**

REFERENCES

- 1 J. Klaessens, T Saris, B Vandeginste, G Kateman, *J Chemometrics*, 2 (1988) 49
- 2 J Klaessens, J van Schalkwijk, P Cox, R Bezemer, B Vandeginste, G Kateman, *J. Chemometrics*, 3 (1988) 81
- 3 P Mijland, J Klaessens, B Vandeginste, G. Kateman, Expert system for the characterization of laboratory sample input signals AUTOCORR, *J. Chemometrics*, (1989) in press
- 4 J Klaessens, L van Beysterveldt, T Saris, B Vandeginste, G Kateman, LABGEN, expert system for knowledge based modelling of analytical laboratories Part I Laboratory organization, *Anal Chim Acta*, .
- 5 T A H M Janse, G Kateman, *Anal Chim Acta*, 159 (1984) 181
- 6 J G Vollenbroek, B G M Vandeginste, *Anal Chim Acta*, 133 (1981) 85
- 7 G M Birtwistle, O J Dahl, B Myhrhaug and K Nygaard, *SIMULA Begn*, Petrocelli/Charter, New York, 1973
- 8 B van de Wijdeven, J Lakeman, J Klaessens, B Vandeginste, G Kateman, *Anal Chim. Acta*, 184 (1986) 151
- 9 J P C Kleijnen, *Statistical Techniques in Simulation*, Dekker, New York, 1975

Part 5

CONCLUSION

CONCLUSIONS AND GENERAL REMARKS

The current task of a LIMS is to control data at the analytical chemistry level. This comprises the functions data acquisition, data storage, report generation and data retrieval. LIMS has only limited capabilities for support at the management level. The most important function at this level is database inquiry. However, in order to arrive at a fully computer supported management in analytical laboratories, additional software modules are necessary that address the management level.

Two modules for this purpose are proposed, FEA for data processing and LABGEN for digital simulation. The most important techniques used for their development are: digital simulation, expert systems, time series analysis and classification based on fuzzy set theory. Both modules are completed and are applicable for a wide range of laboratory organizations. Their structure has been described in detail.

Subsequently, application of FEA and LABGEN to a laboratory for quality control of potable-water is described. The main conclusion is that both modules are flexible enough to be applied in practice. It is important to realize that the modules are developed without bearing in mind any knowledge about the organization of the particular laboratory.

The simulation project was a testcase to validate and optimize the programs. Indeed, some adjustments to LABGEN appeared to be useful. Mostly, these were minor generalizations to allow a more detailed specification of the sample input stream, e.g. with respect to the determination of the tests belonging to a sample. The most comprehensive adjustment was the introduction of the sample pretreatment: a number of tests belonging to one sample are scheduled at the same time (in a parallel way) only after a sample pretreatment has been completed. All introduced adjustments have significance in general and can by no means be considered as an optimization of LABGEN in the direction of the specific application. It is expected that some further adjustments appear to be useful during future applications.

The simulation project was not yet carried out by the laboratory staff. Because the programs require much interaction with the user, it is important to investigate how well they can be handled by the less-experienced user. Attention must be paid to aspects such as: Are the questions posed clear and are the help facilities offered sufficient? Can the user foresee the implications of the answers given? Introduction of LABGEN and FEA into a LIMS environment for use in daily laboratory practice can only be successful if these aspects are sufficiently dealt with. Probably some adjustments will appear to be necessary.

A simulation project consists of three stages: 1) model construction, 2) model validation, and 3) experimentation. During the application of LABGEN which automates the first stage, a subsequent need was felt: the need for automation of or support during both other stages. LABGEN is of great value during the validation stage, because an action in the model can easily be modified without chance of introducing program bugs. The effect of the modification can be measured in a simulation run. However, the ultimate validation question whether the model resembles the real laboratory close enough, cannot be answered by LABGEN.

An expert system for automatic model validation would be useful but is not yet feasible, mainly because for this purpose considerable research is still needed in the area of model validation itself [1]. Such an expert system would also become extremely complex. The first reason for this is the large number of parameters which affect the validation of a model. Secondly, there is a large number of evaluation procedures and tests which can be used. A small advisory expert system would probably be possible if it makes use of the specific properties of the laboratory simulation domain. It must use LABGEN as a starting point, which implies that this research can only be started after the completion of LABGEN.

The same argumentation applies to a system for aid during the experimentation stage. In consultation with the user, this system should propose settings for model parameters to be used for experiments. The proposals can be based on experimental design, a technique which is frequently used for simulation experimentation [2,3]. Subsequently the system should perform the necessary calculations.

Although some further research is useful for application by less-experienced users and for additional support during the simulation project, both modules have proven to be flexible enough to successfully carry out their task in a real laboratory situation. They are of great practical value for management support. Successful incorporation of these modules in the LIMS environment causes the transformation of a LIMS from an information management system into a management support system or Decision Support System (DSS).

REFERENCES

- 1 R.G. Sargent, An exploration of possibilities for expert aids in model validation, in: M.S. Elzas, T.I. Oeren, B.P. Zeigler (eds.), *Modelling and simulation methodology in the artificial intelligence era*, Elsevier, Amsterdam, 1986, p. 279
- 2 B. van de Wijdeven, J. Lakeman, J. Klaessens, B. Vandeginste, G. Kateman, *Anal. Chim. Acta*, 184 (1986) 151
- 3 J.P.C. Kleijnen, *Statistical Techniques in Simulation*, Dekker, New York, 1975

SUMMARY

An important task of the laboratory management is to assure the quality of the information produced by the laboratory. In order to do so, the manager must make decisions at the level of the analysis (precision and accuracy), and at the level of the laboratory organization (costs per analysis, storage of the samples and data, and the delay times of the samples).

This thesis studies how the computer can be used to support the manager during the decision making at the laboratory organization level. The historical laboratory data stored in the Laboratory Information Management System (LIMS) are used as a starting point. This implies that it is investigated which software modules are needed to put these data into effect.

After an introductory first part, the two chapters of part 2 of the thesis describe the major computer techniques that are employed for the development of the software modules. The first technique is digital simulation which is described using a "traditional" application for decision support in a laboratory for liquid chromatography. The second technique is the expert system technique, which is used as a means to provide the elaborate control structures needed.

Part 3 proceeds with the development of the software modules. The first 2 chapters discuss the module FEA (Front End Analysis). FEA consists of an expert system for automatic time series analysis of laboratory input signals, and of 3 programs for data processing. Using an algorithm based on fuzzy set theory, these programs classify the sample entries into a limited number of sample types. The other 2 chapters describe the expert system LABGEN for automatic construction of simulation models: the structure of LABGEN, and the laboratory organization aspects of the models constructed by LABGEN.

After extensive debugging, the modules were successfully applied to a laboratory for quality control. The results of this simulation project are presented in part 4. The modules showed to be flexible enough to handle

the complexities of a real laboratory situation. This part also describes application of the modules to monitor a laboratory in transition.

SAMENVATTING

Een belangrijke taak van het laboratorium management is de kwaliteitsborging van de door het laboratorium geproduceerde informatie. Hiervoor moet de manager beslissingen nemen op het nivo van de analyse (nauwkeurigheid en juistheid) en op het nivo van de laboratorium organisatie (kosten per analyse, opslag van de monsters en van de data, doorlooptijden van de monsters).

In dit proefschrift wordt onderzocht hoe de computer gebruikt kan worden voor beslissingsondersteuning op het nivo van de laboratorium organisatie. De historische laboratorium gegevens, zoals opgeslagen in het Laboratorium Informatie Management Systeem (LIMS), worden als uitgangspunt genomen. Dit houdt in dat onderzocht wordt welke software modules nodig zijn om effectief gebruik te kunnen maken van de gegevens.

Na een eerste inleidende deel worden in de twee hoofdstukken van deel 2 van het proefschrift de belangrijkste computer technieken beschreven die zijn toegepast bij de ontwikkeling van de software modules. De eerste techniek is digitale simulatie, die wordt beschreven aan de hand van een "traditionele" toepassing voor beslissingsondersteuning in een laboratorium voor vloeistof chromatografie. De tweede techniek is die van de expert systemen; deze wordt gebruikt om de noodzakelijke controlestructuren mogelijk te maken.

Deel 3 vervolgt dan men de ontwikkeling van de software modules. De eerste twee hoofdstukken behandelen de module FEA (Front End Analyse). Naast een expert systeem voor automatische tijdreeksanalyse van laboratorium invoer signalen, omvat FEA drie programma's voor data verwerking. Op basis van een algoritme gebaseerd op de theorie der vage verzamelingen classificeren deze programma's de monsterbinnenkomsten in een beperkt aantal monstertypes. De overige twee hoofdstukken gaan in op het expert systeem LABGEN voor de automatische constructie van simulatie modellen: achtereenvolgens de structuur van LABGEN en de laboratorium organisatie aspecten van de modellen zoals geconstrueerd door LABGEN.

Na uitgebreide controle zijn de modules met succes toegepast in een laboratorium voor kwaliteitscontrole. De resultaten van dit simulatie project worden in deel 4 behandeld. De modules bleken voldoende flexibel om een complexe, werkelijke laboratorium situatie aan te kunnen. In dit deel wordt ook toepassing van de modules besproken om een sterk veranderend laboratorium te volgen.

CURRICULUM VITAE

Jo Klaessens werd geboren op 24 februari 1959. Na het behalen van het diploma Gymnasium β in 1977 aan het Bisschoppelijk College te Weert begon hij in hetzelfde jaar met de studie scheikunde aan de Katholieke Universiteit te Nijmegen.

Het kandidaatsexamen scheikunde S2 werd afgelegd in maart 1981. Na een korte periode, waarin hij geschiedenis studeerde aan dezelfde universiteit, vervolgde hij vanaf begin 1982 de scheikunde studie. Het doctoraalexamen werd afgelegd in december 1984 en omvatte het hoofdvak Analytische Chemie (Prof. Drs. G. Kateman), het bijvak Vaste Stof Chemie (Prof. Dr. P. Bennema) en de capita Instrumentele Methoden, Chemische Technologie, Geschiedenis van de Chemie en Quantumchemie.

Deze opleiding werd van november 1984 tot november 1988 gevolgd door een promotieonderzoek bij de vakgroep Analytische Chemie van de Katholieke Universiteit te Nijmegen. In deze periode werd het onderzoek verricht zoals beschreven in dit proefschrift.

Vanaf februari 1988 is hij redacteur van de automatiseringsrubriek van het maandblad Lab ABC.

Vanaf januari 1989 is hij als research medewerker in dienst bij Devtech International B.V. te Heerlen.

STELLINGEN

behorende bij het proefschrift

**TOWARDS COMPUTER SUPPORTED MANAGEMENT OF
ANALYTICAL LABORATORIES**

van

Jo Klaessens

Nijmegen, 18 april 1989

- 1 Verschillende opeenvolgende simulatieprojecten over korte periodes zijn te verkiezen boven een enkel project over een lange periode, zowel met betrekking tot modelvalidatie als met betrekking tot de bruikbaarheid van de resultaten.

Dit proefschrift.

- 2 Zonder beschikbaarheid van laboratoriumgegevens in een LIMS wordt de mogelijkheid van een kosten-effectieve management-ondersteuning m.b.v. simulatie zeer twijfelachtig.

Dit proefschrift.

- 3 De verwachting dat de eerste belangrijke toepassing van expert systemen in de analytische chemie zal liggen op het gebied van informatieverstrekking over analysemethodes voor een gegeven bepaling, gaat voorbij ofwel aan de mogelijkheden van expert systemen ofwel aan de complexiteit van het domein.

H.S. Hertz, Anal. Chem., 60 (1988) 75A

- 4 Er is behoefte aan objectief onderzoek naar het succes van de toepassing in de praktijk van expert systemen.

N.A.B. Gray, Chemometrics Intell. Lab. Syst., 5 (1988) 11

- 5 Het gebruik van negatieve getallen voor de lidmaatschapsfunctie van vage verzamelingen is niet acceptabel en geeft aan dat onvoldoende is nagedacht over het gebruik van de vereniging en doorsnee functies.

A.R. de Monchy, A.R. Forster, J.R. Arretteig, Lan Le, S.N. Deming, Anal. Chem., 60 (1988) 1355A

- 6 Het wegfilteren van lange duur fluctuaties uit een tijdreeks is weinig zinvol voor het bepalen van de bemonsteringsfrequentie op basis van de autocorrelatie functie.

V. Adeberg, K. Doerffel, Fres. Z. Anal. Chem., 327 (1987) 128

- 7 Definiëring van een zich zo snel ontwikkelend vakgebied als chemometrie is eerder van belang uit historisch oogpunt dan uit praktisch oogpunt.

R.R. Meglen, Chemometrics Intell. Lab. Syst., 3 (1988) 17

- 8 Zolang nog lezingen op een congres gewijd kunnen worden aan een algoritme, is het nog niet geschikt voor de praktijk.

- 9 Het zoeken naar een leuke stelling heeft geleid tot een patstelling.