AN EXPERT SYSTEM FOR SHAKING TABLE DIAGNOSTICS

P. TUCKER and K.A. LEWIS

Warren Spring Laboratory, Stevenage, Herts., SG1 2BX, UK
(Received 23 August 1988)

ABSTRACT
An expert system, for fault diagnosis of shaking table operations, is presented. The system is designed as one element of an integrated, computer-based method for plant enhancement, for application in the mineral processing industry. The relationship between expert and conventional software techniques is discussed. The system is based on production rules and has been written in the PROLOG language. Diagnostics are at three levels: the first assessing the known table set-up parameters and the deeper levels providing more detailed analysis in identifying the root cause of the problem. The reasoning, at each level, is developed making reference to fragments of PROLOG code abstracted from the computer program. The system is capable of identifying over 120 faults in tabling operation and has been deployed at two commercial production plants.

Keywords
Expert system; shaking table; diagnostics; heuristics; plant enhancement

INTRODUCTION
With the current uncertainties in the primary raw materials (metals) markets, the need to enhance plant performance and to maintain this improvement in a cost-effective manner is becoming vital for the producing companies. Part of this solution must lie in the use of computer-based tools to aid the process engineer in accomplishing this task. Process modelling and simulation techniques have already demonstrated success in flowsheet design and optimization [eg. 1,2], and material balancing methods are now well accepted in performance analysis. These conventional techniques are not, however, particularly well suited for all applications. They can not, for example, hope to replace the human operator in his response to visual observations (where there is little hard data), and do not implicitly account for equipment malfunction or deterioration. These limitations are particularly acute with diagnostic applications and, for computer techniques to perform well in this role, an alternative strategy must be employed. Expert systems offer this potential. The use of expert systems for diagnosis is now well established in other disciplines [eg. 3-5], and it is in this role that such systems appear to have demonstrated the highest commercial and technical success. The move towards an expert system approach is further motivated here, by its potential to exploit the wide-ranging heuristic knowledge, which is particularly abundant within the mineral processing industry.

The choice of a application was motivated by the strong links between Warren Spring Laboratory, Carnon Consolidated Ltd. (UK), and Beralt Tin and Wolfram SARL (Portugal), who share the common aim of enhancing plant performance through the application of computer methods. Shaking tables are an integral part of processes operated by the two industrial partners and maintaining good tabling performance at these plants is critical in meeting the target grades and recoveries for the final (high grade) concentrates.

Poor table performance can usually be attributed to one, or more, of the following faults:
1. Incorrect table set-up parameters. (stroke, frequency, tilt, wash-water etc.)
2. Problems associated with the feed material. (size distribution, grade, mineral liberation, feed-rate, pulp density, etc)
3. Mechanical problems. (wear and tear etc.)

The aim was to develop an expert system consultant, which would serve (i) to identify tabling faults and inefficiencies and (ii) to advise on the best course of action to take in order to rectify such faults.

KNOWLEDGE ENGINEERING AND SYSTEM DESIGN

A large proportion of the information required to construct the prototype knowledge base was available from in-house 'experts' at Warren Spring Laboratory. In fact, the knowledge engineers were, themselves, minor experts in the domain. Further information was required from published literature, principally [6-8], and from the unpublished PhD thesis of Manser [9]. Some of the information gathered was, however, plainly contradictory and, in such cases, an independent arbitrator was employed. 'Gaps' in knowledge were left to be filled by external expertise at a later stage of the development cycle. This expertise is being provided by the two industrial partners.

The system was targeted at the process metallurgist and plant operator and, for both, a high level of tacit knowledge could be assumed. Much of the knowledge gathered was already formulated in terms of production rules which fitted well into a classic decision tree structure. The diagnostic nature of the problem made it highly suited to goal-driven reasoning, with the structure of the decision tree favouring a PROLOG implementation in particular.

The system was written in PROLOG and implemented on IBM PC computers.

SYSTEM STRUCTURE

Level 1

The system incorporates three levels of diagnostics (Figure 1). The common entry level to the system (level 1) requests fundamental information on the operational set up:

1) Device set up (stroke, frequency, tilts, wash water)
2) Feed characteristics (flowrate, pulp density, particle size)
3) Duty (roughing or cleaning)

Fig.1 Expert system program structure. (The solid lines show the logical path through the system. The dotted lines represent transfer of knowledge to and from the knowledge base)
The user responses are asserted as facts into the knowledge base. Unknowns are treated, in this initial prototype, as the absence of a fact. Quantitative measurements and qualitative judgments (eg. low, medium or high) are equally accepted responses. The level 1 diagnosis tests these input data against the 'rules of thumb' and operating ranges for 'normal' practice. The heuristics (or rules) that are applied are of the form:

\[
\text{fault (FAULT)} :\text{set\_up}(Y).
\]

Here, the functors (fault, setup) describe the condition and the arguments represent a fact relating to that condition. Upper-case arguments represent variables (eg. FAULT is a non-specified fault) whilst lower-case arguments represent constants (eg. fault (washwater, too low) would define the specific fault 'the wash water is too low'). Arguments enclosed within square brackets mean a set of terms. That is setup ([Y]) represents a set of table set-up parameters 'Y'. The symbol ':-' means 'if the facts to the right of the sign are true then the fact to the left of the sign is true'.

The reasoning can, perhaps, be seen more clearly with reference to the examples below, where the setup conditions are specified individually:

1. fault (frequency, too\_low):-frequency (low), feed\_size (fine).
2. fault (stroke, too\_high):-stroke(X),X>40.

Thus, the fault condition 'frequency is too low' is diagnosed when the user indicates his perceived facts that the table is operating at a 'low frequency' and is treating 'fines'. The fault condition 'stroke is too high' is diagnosed if the observed stroke 'X' is greater than 40 mm.

As the heuristics are not infallible, the fault diagnoses, at this level, are only expressed as warnings which the user may override if he feels they contradict his local experience. Contradictions between heuristics do present a problem. For example, roughing operations usually require:

[more water, more ore, less tilt, longer stroke]

whereas fine feed usually require:

[less water, less ore, faster frequency, shorter stroke]

A conflict, thus occurs when a fine feed is treated in a roughing operation. Obviously no general warning can be issued if such a situation occurs. Pointers to the compromise solution will only follow with deeper analysis. A total of 29 potential faults can, however, be identified from application of these simple heuristics. A list of all the likely faults is generated, and reported, using the global search strategy:

\[
\text{Level\_1 :-for all(fault(X),report\_warning(X)).}
\]

Although many of the warnings that are issued might appear obvious (in retrospect) many do represent errors which can easily be overlooked in busy plant environments.

Levels 2 and 3

Whatever their cause, any inefficiencies in table operation will ultimately result in poor product grade, recovery or both. These fundamental aspects are considered in detail within the second diagnostic level. In this level, the system extends the knowledge base by requesting additional, observational data from the user. Unusual behaviour of the mineral bands on the table (and other abnormal visual indicators) are usually good indicators of poor concentration; both usually being different symptoms of the same underlying fault. For example, if the table tilt is set too high: 'FAULT', the mineral concentrate band can narrow significantly: 'OBSERVATION'. This could make accurate cutter setting difficult, resulting either in misplaced concentrate (loss of recovery) or misplaced tailings (loss of concentrate grade): 'CONDITION'. Such features form the basis for further rules within the knowledge base. Following the above example, we have the rule:

\[
\text{fault(tilt,too\_high):-(grade(too\_low);recovery(too\_low)),bands(narrow).}
\]
The visual indications may be 'snapshots' such as bands (wide) or 'historical trends' such as bands (drifting) or recovery (increasing).

Level 2 is built as a decision tree with three primary branches: 'Grade too low', 'Recovery too low' and 'Both too low'. The user is asked to specify one of these primary inefficiency conditions in order to initiate the level 2 analysis. The given response is asserted as a fact into the knowledge base.

If the user has no information regarding the above efficiencies, diagnosis can still proceed but using the observational data only (level 3 analysis). On entry to level 3, it is not known a priori whether the observations are actually indicative of a fault; a potential fault situation only becoming positively established by its effect on table efficiency. Observations can, however, provide the basis from which inferences about the likely efficiencies can be made. These inferences will include the three primary error conditions of level 2 plus the conditions 'Grade and recovery appear satisfactory' and 'Grade and recovery appear satisfactory but operational difficulties could occur'. However, because less information is available at level 3 than at level 2, the system can only draw on a subset of the rule-base with consequence that less faults can be diagnosed. For example, in the example given above, the fault: fault(tilt, too_high) can not be diagnosed, at level 3, because the conditions: grade(too_low) or recovery(too_low) have not been established as facts.

The essential procedural differences between levels 2 and 3 can be summarised as follows:

**Level 2:**
```
fault(FAULT):- condition(X), set_up([Y]), observation([Z]).
```

**Level 3:**
```
condition(X):- set_up([Y]), observation([Z]).
fault(FAULT):- set_up([Y]), observation([Z]).
```

In plainer English, the first set of expressions states that a fault 'FAULT' is identified which will account for the given inefficiency condition 'X', consistent with the given set up parameters '[Y]' and the given set of observations '[Z]'. The second set of expressions states that the efficiency condition 'X' is deduced from '[Y]' and '[Z]' alone and that a fault 'FAULT' may also be deduced from the same information.

The final diagnostic report is either expressed as a single error eg. FAULT="tilt too high" or as a conjunction of two errors eg. FAULT="tilt too high and wash water too high". At the time of writing, in excess of 90 different faults can be identified at these levels. Sometimes, of course, the system will fail to identify a fault, either because of insufficient user data or missing rules or because there actually is no fault. In these cases, a no-fault diagnosis will be reported. i.e.:

```
level_2:(not fault(X)), report("No fault can be identified").
```

### Additional details

Whilst Level 1 simply uses the inference mechanism which is implicit within PROLOG (i.e. backward-chaining, 'goal-driven' reasoning), a substantial procedural element must be built into the deeper levels in order to maximise the efficiency of the more-complex diagnoses. The search routes have been designed such that the major faults are investigated first and, within each search, the questions ordered for economy of user interaction. Each fault is established as true when its set of contributory facts is proven. The facts are tested sequentially; the order designed to minimise redundancy. This imposes a forward-chaining, 'data-driven' executive on the program at these levels. An example will be given later.

A second design feature refers to the set-up parameters, asserted at Level 1. In the subsequent levels, these facts serve only to confirm but not, on their own, to refute the diagnoses. For example, the visual observations may indicate that the stroke was 'too high'. If the user has stated that the stroke was 'high' then the diagnosis is confirmed. If, however, he stated that the stroke was 'low' then a conflict arises. This could, either (i) pinpoint a misconception of how the table is actually set up or (ii) may be indicative of
a mechanical fault in the stroke mechanism (i.e. another diagnosis). To differentiate between these two possibilities, further data may be required. This procedure is illustrated below:

\[
\text{fault(stroke,too_low)}: -\text{observations([X]), stroke(low)}. \\
\text{conflict(stroke,too_low)}: -\text{observations([X]), (not stroke(low)}. \\
\text{fault(mechanism,faulty)}: -\text{conflict(stroke,too_low), further_data([Y])}. \\
\text{level_3} :-\text{(fault(FAULT), report_fault(FAULT), (conflict(FAULT), report_conflict (FAULT)).}
\]

Thus, the best guess solution 'the stroke is probably too low but this conflicts with what you said earlier' will be issued when the two definite fault diagnoses both fail. The economy of the solution is illustrated in that the further data (which may also require further questioning of the user) is not tested if the first fault diagnosis is confirmed.

During the consultation, the expert system design also allows for subsidiary warnings to be issued where appropriate. These warnings will not relate to prime faults but to possible contributory factors. For example, visual observations could indicate that slimes were adhering to the table surface. The true fault may be 'wash water too low', but in reaching this conclusion, the system might first realise the subsidiary goal 'pH may be too low'. This would constitute a warning. Warnings are incorporated into the search path as follows:

\[
\text{fault(wash_water,too_low)}: -\text{slimes_adhering, warning_message(pH_TOO_LOW), further_data([Y])}. \\
\]

Thus, once the fact 'slimes adhering' has been established, the warning message will be issued immediately. The warning is not suppressed if the further data fails to support the fault diagnosis, the applicability of the warning being solely dependent on the fact that the slimes are adhering to the deck.

**IMPLICATIONS AND CONCLUSIONS**

The expert system analysis allows for periodic or responsive monitoring to be undertaken without calling on expensive, and time consuming, analytical procedures. It allows the metallurgist and operator to check if any unusual 'quirks' will have serious implications on performances, and will delineate potential problem for further attention. The system should identify the most likely cause of the problem and should advise on the most fruitful line of action (to eliminate its root cause rather than just to alleviate its symptoms). The expert system can not, however, quantify the course of remedial action. Process modelling and simulation techniques must be used here, to reset the table to its optimum performance. This dual approach, utilising simulation for process planning and expert systems for performance analysis, should provide the plant metallurgist with powerful tools for promoting and maintaining plant operation. The methodology is illustrated schematically in Figure 2.

The expert system reported here is continually expanding, as more rules are established and 'missing' knowledge is acquired. The current system comprises over 300 rules and is capable of detecting in excess of 120 likely faults. The system has been installed at Carnon Consolidated Ltd. and at Beralt Tin and Wolfram SARL, where its full potential is being evaluated. Local production rules, incorporated from the combined experience of the two plants are expected to considerably enhance the effectiveness of the system.

**ACKNOWLEDGEMENTS**

The contributions from Carnon Consolidated Ltd. and Beralt Tin and Wolfram SARL are gratefully acknowledged. The work was supported by the Department of Trade and Industry and through the Commission of the European Communities Third Raw Materials Program.
Fig. 2 Expert systems as part of an integrated methodology for plant enhancement

REFERENCES