UNCLASSIFIED

DTIC FILE COPY

AD-A228 454

AR-005-997

Communications Division

REPORT
ERL-0509-RE

XSHELL - A GENERAL PURPOSE EXPERT SYSTEM SHELL

by

Allan J. Mack

DTIC
S ELECTE D
NOV
C6 E

APPROVED FOR PUBLIC RELEASE

ELECTRONICS RESEARCH LABORATORY

2

AR-005-997

DTIC
ELECTE
NOV 13 1990
S E D

DSTO
AUSTRALIA

ELECTRONICS RESEARCH LABORATORY

THE UNITED STATES NATIONAL
TECHNICAL INFORMATION SERVICE
IS AUTHORISED TO
REPRODUCE AND SELL THIS REPORT

Communications Division

REPORT
ERL-0509-RE

XSHELL - A GENERAL PURPOSE EXPERT SYSTEM SHELL

by

Allan J. Mack

SUMMARY

Available expert systems show some significant deficiencies when applied to practical applications requiring reliability and maintainability. This report describes an expert system shell, XSHELL, which addresses these problems and which has been successfully used in the Intelligent Frequency Management System.

© COMMONWEALTH OF AUSTRALIA 1990

MAY 90 COPY No.

APPROVED FOR PUBLIC RELEASE

POSTAL ADDRESS: Director, Electronics Research Laboratory, PO Box 1600, Salisbury, South Australia, 5108.

This work is Copyright. Apart from any use as permitted under the Copyright Act 1968, no part may be reproduced by any process without written permission from the Director, Publishing and Marketing, Australian Government Publishing Service (AGPS).

Enquiries should be directed to the Manager, AGPS Press, Australian Government Publishing Service, GPO Box 84, Canberra ACT 2601.

Accession For

NTIS GRA&I	☒
DTIC TAB	□
Unannounced	□
Justification	

By

Distribution/

Availability Codes

Dist	Avail and/or Special
A-1	

CONTENTS

APPENDICES

1 INTRODUCTION

An expert system shell is a program which is ideally independent of the domain (subject) of application. A problem is represented by facts and rules which the shell processes as data. Although it may be large, the shell is a conventional procedural program.

The shell described in this paper is the result of a study of existing shells and of the needs of the Intelligent Frequency Management System. The outcome is a comprehensive tool for the development, execution and maintenance of rule-based expert systems. Several unique features can ensure a high level of completeness, reliability and maintainability in the application.

The degree of detail which follows is provided not only as a guide to users but also as a reference for the assessment or specification of other shells. This issue refers to XSHELL version 2.11.

Expert systems usually divide the rules and facts into the domain, which describes the general case, and the context, which is the part that can vary from one problem to another.

In XSHELL, facts and rules describe the problem domain. The assignment of certainties and values to the facts describes the context. The solution of problems is goal driven using a combination of backward and forward chaining.

Each fact is described by a text. Associated with each fact is either a set of descriptive states each having a probability (certainty), or a numeric value (or array of values). I refer to the descriptive facts as being 'qualified' and the numeric facts as 'quantified'.

Rules consist of fact tests (the IF or condition part) and fact assignments (the THEN or consequence part).

2 NOTATIONAL CONVENTIONS

In this paper, formats are described in which square brackets ([...]) indicate an optional term. If the option consists of alternatives, a vertical bar separates them ([...|...]). Curly brackets with a vertical bar separator ({...|...}) indicates alternatives, one of which is required.

3 QUALIFIED FACTS

A qualified fact appears as a statement of a truth in natural language (say English). The preferred form of the fact is 'subject verb qualifier', where 'qualifier' is a noun, adjective, adverb or equivalent phrase. In the example 'the error rate is moderate', 'moderate' is the qualifier, perhaps one of several ranging from 'very low' to 'very high'.

Unlike some systems, XSHELL accepts space characters as parts of the descriptive texts.

Each fact has several possible qualifiers. In the definition of a fact XSHELL requires the designer to name the fact (the subject-verb) and to define and name each of the qualifiers. Each qualifier has an associated certainty expressed as a percentage (0 to 100). A feature of XSHELL is that the certainty of a fact can be distributed amongst several or all of its qualifiers. The certainties taken collectively describe the condition of the fact. The certainties cannot total more than 100%, and so one qualifier is emphasised at the expense of others. Rules can exploit this implied EXCLUSIVE-OR function.

This is the structure of a qualified fact:

FACT NAME	QUALIFIERS	CERTAINTIES
SUBJECT AND VERB	TEXT 1	CERTAINTY 1
	TEXT 2	CERTAINTY 2
	:	:
	:	:
	TEXT n	CERTAINTY n
		TOTAL<=100%

This is an example of a qualified fact:

FACT NAME	QUALIFIERS	CERTAINTIES
The signal to noise ratio is	very low	0
	low	90
	moderate	10
	high	0
	very high	0

XSHELL considers a fact to be undefined until the program provides certainties, either by keyboard entry, file input or rule execution.

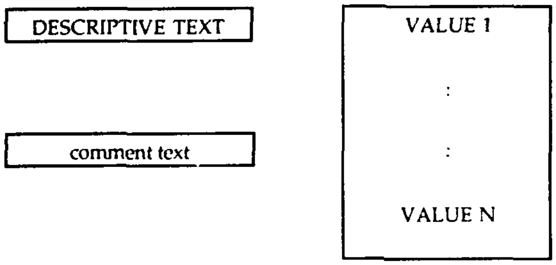
Where certainties total less than 100, XSHELL shows the shortfall as 'unknown'.

4 QUANTIFIED FACTS

Quantified facts can be simple numeric variables or arrays. Numeric quantities are often acquired from external systems and so may have rather cryptic names. There is therefore provision for an accompanying explanatory or comment text.

The value(s) are stored as single precision numbers. Rules can assign values to these facts, use them in expressions and test them for their value. Like qualified facts they can be flagged as undefined.

A quantified fact has the following structure:



XSHELL rules can refer to whole arrays or to specific members of an array. References to whole arrays imply an iteration through all the members of the array, which XSHELL executes automatically. For example, you can test each member of one array against some condition and assign a result to each member of another array, in only one rule. The members of the second array can therefore vary according to the members of the first array. When XSHELL executes the rule it recognises the "each" requirement and repeats the execution from the first to the last member. When you construct such a rule, XSHELL insists that the arrays are of the same size.

There is a subset of quantified facts being inbuilt functions. These include read-only time and date functions, from seconds to years, derived from the computer clock. They are sampled and held at the beginning of a goal solution run.

There is also an inbuilt read-only function which returns the value of the subscript during array iteration. You can therefore affect each member of an array by its position.

Further details on the use of arrays are given in Appendix A.

Functions are available for the derivation of array parameters such as maximum, minimum, and also many statistical functions (see Appendix B).

When facts are created, XSHELL ensures qualitative facts are lower-case and quantitative facts are upper-case. To distinguish them further the latter are enclosed in square brackets.

5 RULE STRUCTURE

Each rule consists of a condition (IF) part and a consequence (THEN) part. This is the structure of a rule:

	CONDITION PART			CONSEQUENCE PART	
IF	CONDITION 1	AND	THEN	CONSEQUENCE 1	AND
	CONDITION 2	AND		CONSEQUENCE 2	AND
	:			:	
	:			:	
	CONDITION N			CONSEQUENCE M	

comment text

5.1 The IF part

The IF part can contain any mix of qualitative and quantitative tests. The lines of tests are linked by AND function operators.

- Each test of a qualified fact has the following format:

  IF factname [NOT] [FROM] qualifier1 [TO qualifier2] THEN

  The *optional FROM...TO qualifier2* indicates a range of qualifiers. In rule execution XSHELL takes the sum of the certainties associated with the qualifiers in the range. This sum is the equivalent of an EXCLUSIVE-OR.

  The simple and common form of a test is:

  IF factname qualifier THEN

- A quantified fact appears in the IF part in the following format:

  factname relational_operator mathematical_expression

  The relational operator can be =, <>, >=, >, <=, or <.

In rule execution XSHELL compares the value of the fact with the expression according to the operator and assigns a certainty or zero according to the truth of the relationship. The certainty is the product of the certainties of the terms in the line.

5.2 The THEN part

The THEN part can contain any mix of qualitative or quantitative assignments.

- *Each qualitative assignment consists of a fact, one of its qualifiers and a* certainty to be assigned.

- A quantitative assignment allows the value of a mathematical expression to be assigned to a quantitative fact. The expression is limited in form for both IF and THEN parts, thus:

  [math_funct] {constant | factname1} [math_op {constant | factname2}]

  The maths functions available are described in Appendix B.

  The maths operator can be +, -, *, /, MOD or exponentiation.

  XSHELL permits only quantitative facts to appear in an expression. The *constants can* be assigned floating point numbers.

When XSHELL displays the assignments they are shown with AND conjunctions. These conjunctions are not logical but simply express in English the multiplicity of the assignments. For example:

> IF the signal to noise ratio is low AND
> the doppler shift is small THEN
> the bit error rate is low (80) AND
> transmission is recommended (100)

Multiple assignments are those which depend upon the same (AND) conditions. However the degree of dependence is defined by separate assigned certainties and the facts may have other rules contributing to them (see Inter-rule Processing below).

Each rule may have an associated explanatory text.

6 RULE AND FACT ORGANISATION

XSHELL assigns an index (integer number) to each fact, each fact qualifier and each rule. Furthermore, so that rules may accommodate various numbers of IF facts and THEN facts, there are intermediate reference tables so that rule arrays do not have to allow for the maximum number of facts in each rule. Instead, the facts are stored contiguously in one array. This removes the practical limit on the number of facts of each type in any one rule without wasting memory.

Another intermediate reference table allows each fact likewise to have any number of qualifiers provided total storage is not exceeded. Rule searches and rule evaluation therefore require two or three levels of nested array indexing. These intermediate arrays are invisible to the user.

XSHELL uses text strings (describing each fact and qualifier) only in the presentation of information to the user, and the user types each text only once, at definition. All subsequent user references to facts and qualifiers are by their index numbers. This includes the construction of rules. The user always has access to lists of the texts with their associated indices. These lists are limited by context. For example you may choose only quantitative facts when you are constructing a maths expression.

Apart from the advantages of higher processing speed and fewer user keystrokes, the avoidance of text processing within XSHELL eliminates ambiguity and any need for language interpretation or processing.

There is of course no need for XSHELL to store the words IF, THEN, NOT and AND with the rules. It adds these only during display.

7 INTRA-RULE PROCESSING

The rule structure of XSHELL provides only for the logical AND combination of facts and the EXCLUSIVE-OR of their qualifiers in the IF (condition) part. Independent AND involves the multiplication of certainties and the multiplication is commutative. Each qualifier can be negated (NOT) in which case XSHELL uses 100 minus the qualifier's certainty (or range of certainties).

The THEN part of a rule consists of certainty or numeric assignments (third and fourth lines of the above example).

Each certainty assignment consists of a fact, one of its qualifiers and a value of certainty to be assigned. This certainty represents an assessment by the rule designer of the extent to which the qualifier is affirmed by the condition part of the rule. XSHELL multiplies the certainty specified in the THEN part of the rule ('80' in line 3) with that derived for the whole of the condition part to obtain the certainty to be assigned to the qualifier. This certainty, after combination with those from other rules defining the same fact, would then be used in the rule(s) where this fact and qualifier appears in the condition part.

Quantitative facts are assigned the value of the RH expression provided that the condition certainty exceeds that obtained for the same fact in any earlier rule executed.

The values input to expressions may have certainties less than 100% if they have been determined by other qualitative rules. In the absence of any knowledge of the statistical distribution of the values, XSHELL can only multiply the condition certainty with those of term 1 and term 2, assuming 100% for constants.

It is advisable, of course, not to derive numerical values from uncertain logical states. If you find it to be the preferred method, you should take care to address the distribution of values through a suitable set of rules.

8 INTER-RULE PROCESSING

In XSHELL the existence of more than one rule defining a qualitative fact does not constitute a conflict. Rather, XSHELL regards each rule as contributing positively to the fact. So XSHELL combines the certainties with an INCLUSIVE-OR function (the conventional OR function). If two certainties to be ORed are C1 and C2 then, using a 0 to 1 scale, by inspection the OR function is:

$$C1 + C2 * (1 - C1).$$

Alternatively we can use De Morgan's rule on AND (multiplication), giving:

$$1 - ((1 - C1) * (1 - C2)).$$

Of course, these simplify to the same form:

$$C1 + C2 - C1 * C2.$$

Note that this expression is independent of the order in which the terms are combined - that is, the OR is commutative. This can be extended to say that any number of certainties can be combined in any order with the same result. (Try multiplying out a succession of certainty combinations.) The indeterminate order of rule execution relies on commutation to produce deterministic results.

XSHELL presets each fact's certainties to zero immediately prior to the evaluation of the first rule found defining the fact. This preserves the certainties of facts not defined by any rule until changed by user action.

XSHELL does not reset the value of a quantitative fact at the first effective rule, only its certainty. Resetting the certainties ensures that the first calculated value is effective. You can exploit this feature in that a quantitative fact can appear in each side of an as^signment such that X = f(X). The evaluation of this construction is dependent on the detection of rule looping, explained below.

After all the rules defining a qualitative fact have been evaluated, XSHELL adds the fact's certainties and, if they exceed 100, it reduces them all in proportion.

XSHELL resolves conflicts of numeric assignments oy using the one with the highest certainty.

All certainties are maintained as integer percentages and products include 0.5% rounding.

9 CERTAINTY AUGMENTATION

The OR function allows the rule designer to augment the output of a rule with the output of another. For example, suppose we have a set of rules which determine the likelihood of rain based on changes in wind direction, barometric pressure and temperature. An additional rule could state that if the humidity is high there is an independent additional certainty of rain of 20%. That is, rain is 20% more likely. The OR algorithm combines this value with the previously derived certainty, increasing it. But the rule does not say that if the humidity is high then rain is 80% (that is, 100-20) unlikely.

10 THE SOLUTION OF GOALS

XSHELL uses a combination of backward and forward chaining. Rather than starting from a proposition, the user nominates a fact as a goal, in effect asking "What is the state/value of this fact?". XSHELL evaluates that fact by alternatively backward chaining to find the relevant rules and input facts, and forward chaining, evaluating rules.

While backward chaining, each time XSHELL finds a fact amongst the rules, it pushes on to a stack the indices of the old fact and rule. The search algorithm repeats for each new rule found.

When all the conditions for a given rule have been derived XSHELL evaluates it. Whenever a rule is evaluated (or skipped - see below) a fact and rule are popped from the stack and the se rch continues for that fact. When XSHELL has evaluated all the rules for a fact, it then considers the next fact in the popped rule.

Note that mathematical expressions in either the IF or THEN parts of a rule can contain quantitative facts which, being 'inputs' to the rule, must be sought in the same way.

XSHELL works through the tree structure of the relevant rules, alternating between backward and forward chaining, with the stack pointer effectively running up and down each branch of the tree. At any time the stack contains the path from the rule currently being considered down to the goal, so that where there are multiple outcomes in a rule, the required one is followed.

In backward chaining, ultimately XSHELL finds facts for which there are no rules. Expert system shells vary in where they proceed from here. XSHELL looks first to a fact's certainties in the context and uses those if they are defined, otherwise it asks the user to provide them.

Although the certainty of only one qualifier may be required at a time (the qualifier in the condition), XSHELL expects the user to enter values for each possible qualifier for that fact. Other rules may require some of the other qualifiers and entering them all together is more likely to result in a consistent set of values. Operator entries are preserved at least until all goal qualifiers are done.

When the certainties of a fact entered by the user total 100%, XSHELL seeks no more and sets the remaining certainties to zero. Operators can choose a single qualifier to be 100% by nominating it, or they can work down the list distributing the 100% among one or more qualifiers.

In the solution of goals, XSHELL distinguishes between undefined and unknown facts. A user can declare that a fact's state is not known, thereby defining it. If XSHELL subsequently encounters the fact, it will not repeat the user query but will accept that the fact was declared unknown

The INCLUSIVE-OR function implies a significant principle in XSHELL's goal solution algorithm, namely that all rules defining each required fact must be sought and the required facts in turn are those appearing as inputs to each rule found.

10.1 Loop detection

A rule output can be required by more than one other rule. The property of the OR function would result in a modification of the output certainties if the rule were re-evaluated. In addition there could be significant computation involved in the structure above the rule.

Therefore each rule has an associated counter which records the number of times it has been encountered. This not only prevents re-evaluation but also detects looping generally. As some looping is legitimate, only a simple and large count threshold is needed to detect indefinite loops. XSHELL uses the total number of qualifiers of all the facts as the threshold. If the looping exceeds the threshold XSHELL aborts the solution, issuing a diagnostic.

10.2 The 'why' option

The existence of the stack path to the goal makes a 'why' option easy to incorporate. When XSHELL asks users to supply certainties or values, they can ask 'why' it needs them. XSHELL will display the rule in which the input fact occurs. Repeating the 'why' shows successive rules taken from the stack toward the goal until the goal is reached. Further questioning repeats the path. At any time, the original request for certainties or values can be answered.

10.3 Rule skipping

If the user declines to supply certainties or values, XSHELL ceases considering that rule and any other rule using that fact. The user is effectively defining the fact as unknown. This is better than using zero certainties, as some systems do, especially where a NOT operator is present which would give an effective 100% certainty.

The skipping of rules propagates down towards the goal. The skipping ceases where a potentially undefined fact is defined by an alternative rule (by the implicit OR function). The extent to which the undefined fact affects the goal's certainties obviously depends on the scope of the rules, the facts they invoke and the certainties they assign.

A rule is also skipped if the accumulating condition certainty becomes zero. It cannot then become greater than zero because IF certainties are multiplied (AND).

10.4 Fact skipping

There are two situations where XSHELL will terminate prematurely the search for a rule's input facts. The first occurs when the user declines to provide certainties. The rule(s) for which they are required cannot be evaluated and so any remaining inputs are irrelevant.

The second occurs when the IF certainty becomes zero. There are two such cases, namely, when the fact certainty is zero, or when the certainty is 100% and a NOT operator precedes it.

This can have effects which may puzzle users. First, they may wonder why a particular fact was not requested. Second, where the fact may still be required by another rule, they may wonder why the order in which XSHELL seeks facts changes from one run to another.

On the other hand the rule designer can exploit this feature to minimize user entries. XSHELL seeks input facts in their order in the rule, and so when rules are constructed, the designer can place a controlling condition, which is more likely to have values of 0 and 100, as the first in the rule.

11 GOAL REASONS

When XSHELL displays the results for a goal it optionally displays a panel containing one of the rules which directly affected the goal. Other effective rules can be displayed by scrolling, using the page-down and page-up keys. The only rules that are accessible are those which contributed to the goal, that is, those whose condition certainty exceeded zero.

Each displayed rule shows one of its input facts highlighted. This fact, showing its value or state, is listed below the rule. You can display the rule affecting this fact with the left-arrow key. You can scroll through the input facts of a given rule with the down-arrow and up-arrow keys, and so display a rule for any chosen input fact.

Each displayed rule also highlights the output fact which was used to select the rule. The right-arrow key leads you to the same rule using this fact from which you moved with the left-arrow, regardless of any interim scrolling that you may have done. Using a combination of these keys you can examine any of the rules and facts affecting the goal.

At any time you may press the enter key to leave the results and reasons panel.

12 GOAL COMPLETENESS AND RELIABILITY

A common criticism of the use of expert systems is the difficulty of assessing the completeness of the rules defining the goal. XSHELL has several features which allow the designer to assess the quality of the rule-base, the input facts and the results.

In a large rule-base it may not be practical to ensure that all the qualifiers' certainties are independently complementary under all conditions. That is, after all goal qualifiers have been evaluated, the certainties may total more than 100%. In this case XSHELL reduces each one in proportion. In trace mode (see below) this reduction is reported. A certainty total of 100% does not guarantee completeness in the rules, but a total of less than 100% does suggest incompleteness.

The following summarize the indications of goal quality:

12.1 Undefined goal

This is fundamental. XSHELL indicates the complete absence of rules defining a goal.

12.2 The 'unknown' indicator

If a goal shows some 'unknown' certainty then either there are relevant rules missing or a fact has been incompletely defined (its certainties do not total 100%). XSHELL issues an appropriate diagnostic if the unknown exceeds 15%.

12.3 Looping

The trace option reveals all rule looping. Without the trace, XSHELL diagnoses looping only if it becomes excessive, in which case the scan is aborted.

12.4 Goal reasons

This option already described allows the designer to review the effective facts and rules, following the rule structure.

12.5 Goal solution log

This option displays lists of effective facts and rules in order of their index. Note that facts not needed because of rule skipping are not shown.

13 DIVERSITY OF FACT QUALIFIERS

Many facts are essentially bipolar (two-qualifier). That is they have two extremes, and their state can be defined by a complementary pair of qualifiers.

However, a more reliable indication of the qualifier may be obtained from the user by presenting several qualifiers which represent shades of meaning. The user can then place a high certainty against the best fit, rather than doing a mental calculation and distributing the certainties between two extremes. The penalty is that additional rules are required to cover these several input qualifiers.

But the derived facts, whether intermediate or goal, may only need to be bipolar, and the rules, if correctly constructed, will distribute the certainties between the two qualifiers appropriately. For example, the user may be presented with a choice between excellent, very good, good, fair, poor, bad and very bad, but the goal may need to be simply acceptable or unacceptable, with the accompanying certainties indicating a figure of merit.

Other truly multiple qualifier facts cannot be dealt with this way, for example, a selection of colours. Rather, quite different intermediate facts may be derived from the various input quaiifiers. In this case the qualifiers must be carefully checked to confirm that they are mutually exclusive, since the user cannot emphasise one qualifier except at the expense of the others. If they are not so, separate facts should be used.

14 BINARY LOGIC

The designer can construct a two-state (true/false) logic system by using certainties of only 0 and 100%, both for facts and in rule assignments, and by giving each fact two qualifiers, say, true and false, acceptable and unacceptable, etc.

15 TESTING AND DIAGNOSTICS

Once a fairly comprehensive set of rules has been established it can be tempting to place undue trust in the results. Expert systems can produce results which consistently appear 'about right' but which have some bias due to missing rules or incorrectly judged certainty assignments. As with any software it requires some discipline, often with help from someone independent, to test the logic extensively.

XSHELL has several features with which the designer can assess the integrity of the application.

- The menu option to save the domain to a file also tests for rule completeness. XSHELL lists those logical facts whose states are not all used by the rules. The unused states are indicated.

- The trace shows each fact sought, how often each rule is found, the stack level at each rule, the evaluation of expressions, the results of each rule evaluation, the certainties of facts taken from the facts-base, skipped facts and skipped rules. The trace can be turned on at the beginning of a manual run, or triggered at a nominated fact whenever it is encountered. This trigger option allows XSHELL to run fast until an area of interest is reached. The trace can be turned off (ESC) during goal execution and fast execution resumes. If the trigger option was chosen, the trace will resume at subsequent encounters of the same fact.

- At the completion of the run there is a log option. The log lists the facts and rules contributing to the solution. In automatic mode (see below), the log is written to a file provided the REASONS option is not selected (also see below).

- There are diagnostics for all overflow conditions, but user selections exceeding the number of rules, facts or qualifiers, result in re-prompting.

- The times at which a goal solution run both starts and finishes are displayed.

- Missing or corrupt input files are diagnosed.

16 MATHS ERRORS

The mathematical functions include error traps as follows:

LOG of a number, < = 0 returns -1E+38
TAN overflow returns 1E+38
SQR of a negative number returns SQR(ABS(number))
x MOD 0 returns 0.

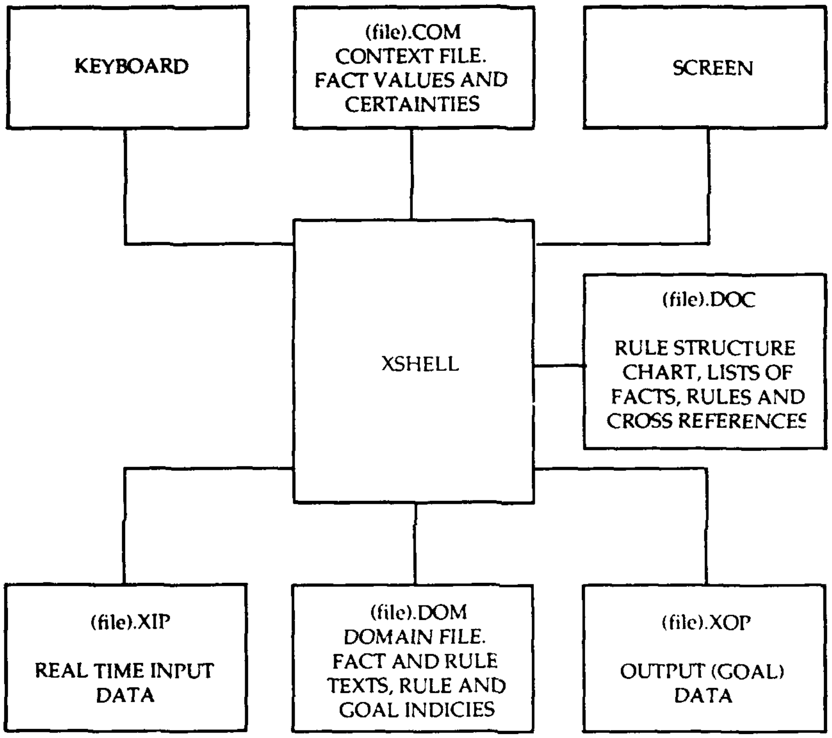
The mathematical operators return ±1E+38 for overflow according to the sign of the operands.

17 MULTIPLE GOALS

As with facts generally, the qualifiers of a goal are complementary and so one goal may not adequately represent the required solution. XSHELL therefore provides for the solution of many goals in the one run. Operator inputs and evaluated rules for any one goal remain effective for subsequent goals, and so all the solutions are for the one set of conditions.

18 SYSTEM STRUCTURE

The diagram shows the relationship of XSHELL to the data that it processes:


KEYBOARD
(file).COM CONTEXT FILE. FACT VALUES AND CERTAINTIES
SCREEN
XSHELL
(file).DOC
RULE STRUCTURE CHART, LISTS OF FACTS, RULES AND CROSS REFERENCES
(file).XIP
REAL TIME INPUT DATA
(file).DOM DOMAIN FILE. FACT AND RULE TEXTS, RULE AND GOAL INDICIES
(file).XOP
OUTPUT (GOAL) DATA

19 FILE I/O

So that XSHELL can operate as a real time processor or controller, data can be read from and written to files accessed in turn by other applications. At the beginning of goal solution, XSHELL reads data from the input file (xxxxx.XIP), if it exists, where xxxxx is the name of the expert system. After the solution of each goal, XSHELL writes the goal to the output file (xxxxx.XOP).

20 FILE FORMATS

In the run-time I/O files, the format of qualitative facts is:

> fact_text_name qualifier_text_name [(certainty)]
>
> On input, if the certainty is omitted, 100% is assumed.

and the format for quantitative facts is:

> [fact_text_name] value
>
> where value is a floating point or integer constant. XSHELL diagnoses fact-texts that it cannot match.

The documentation file (xxxxx.DOC) consists of rule structure charts drawn with printer characters, and lists of facts and rules in a text form (see below).

The domain (xxxxx.DOM) and context (xxxxx.CON) files contain lists of text strings and indices for the multiple level internal arrays, which are not readily followed. It should never be necessary to be concerned with these formats, since the data need be created and used only by XSHELL. Editing rules and facts should be done only using XSHELL.

All files are in ASCII code.

21 DATA PRESETS

XSHELL resets the certainties of facts at the first effective rule found, so that if it does not find a rule it can use any preset certainties. XSHELL resets all the facts first affected by the rule.

There are two side effects of this to remember:

- One consequence is that any facts not affected by the run at all will retain their old certainties.
- The other consequence is that some facts, being affected by rules but not required by the goal, may be only partially evaluated.

Therefore the user cannot examine the certainties in the facts base after a goal solution and assume that they all apply to the current context, without having some knowledge of the rules. Of course, any fact specifically required should be made a goal.

Prior to the first goal solution, that is, at the beginning of a solution run, XSHELL resets all the flags indicating that the user has declined to supply certainties and it also resets the rule loop counters.

22 DOCUMENTATION GENERATION

As an aid to the design of reliable expert systems, the user can command XSHELL to generate a documentation file based on the facts, rules and goals. The documentation consists of:

- A rule structure chart for each goal, by rule index. Repeated rules are enclosed in brackets and the structure above them is not repeated.
- A numbered list of the fact and qualifier texts.
- A numbered list of the rules in text form.
- A fact-rule cross reference table by fact text and rule index.
- A list of unused rules by index.

Whereas in goal solution XSHELL seeks all rules for all fact states, this algorithm produces rather extensive charts even for small rulebases. The chart option seeks only those rules that define specific states or ranges of states. Only if the NOT operator is present in the rule are all states sought.

The documentation option also allows you to sort the rules into the order of their comment te. ts. This allows you to group your rules according to sub-functions of your choice, and is particularly useful if you add new rules. You could begin each comment with a certain key word for each group, or perhaps you could use Development Documentation System (DDS) F-numbers. If you use a numbering system, remember that the text sort does not resolve missing leading blanks. For example "12" will precede "2", but will not precede "02" or " 2".

23 OPERATING XSHELL

The general form of the DOS call to XSHELL is:

```
XSHELL [filename [RUN/TEST] [REASONS]]
```

where filename has no extension. TEST is the default.

XSHELL has two principle modes of operation - RUN and TEST

23.1 Manual (or Test) mode

This is the mode in which the domain is created and maintained. If the filename is omitted XSHELL will ask for it.

The user is responsible for loading the domain and context files. Prior to loading, the menu appears similar to:

```
 0 facts          0 rules         41480 free text
Facts            Rules              Control
0 List           5 List             F1 Help
1 Create         6 Create
2 Modify         7 Modify           F2 Run
3 Delete         8 Delete
4 Search         9 Search           F7 Sort/Doc
Domain           Context
                                    F8 Colour
F3 Load          F5 Load            F9 Set goals
F4 Save          F6 Save            F10 Quit
```

After the domain is loaded or created, the numbers of rules and facts are displayed.

The menu is displayed in co'our, highlighting the headings and generally improving its readability over this monochrome representation. Several colour combinations are available and the one selected is saved with the domain.

The commands provide for:

- Load and save domain and context files.
- List, create, modify, search and delete facts and rules.
- Create and solve goals with optional trace/log.
- Sort rules, generate documentation.
- Request help, change colour and quit.

The function (F) keys are used as single key commands. Other commands which are followed by fact or rule numbers can be chained with a dot separator. For example, 7.3 means modify rule 3. These commands are terminated by the ENTER key.

The list commands can specify all (default), a single fact or rule, or a starting number. For example 0.4- means list from fact 4 onwards.

Most operations can be aborted using the ESC key. ESC during trace turns trace off, but ESC at a request for fact input aborts the goal run.

Fact searches are done by matching an operator specified text string. Rule searches are for facts specified by their index.

If the last panel displayed extends below the top of the menu, XSHELL will require an extra ENTER key to display the menu, so that the display is not hidden. Even so, the menu panel can be toggled on and off with CTRL-0.

The Quit option checks for modifications since the last save and prompts the user if necessary.

23.2 Automatic mode

This mode is invoked by including the word 'RUN' in the DOS command.

In this mode the user has no access to the menu commands and so cannot modify the domain.

XSHELL loads the domain and context files, solves the goals (saved in the domain file) and exits to the operating system. All the values and certainties required can be in the context and data input files, so that no user action is required. Otherwise XSHELL will ask the user to supply them as it meets them.

23.3 Keyboard line editor

Whenever XSHELL requires user input it invokes a line editor. The editor is permanently in insert mode, so that if you left shift (with the left-arrow), further entries are inserted. The backspace key is the complement of the forward-space (spacebar). The delete key is also functional.

BIBLIOGRAPHY

Bramer, M.A.	"A Survey & Critical Review of Expert Systems Research", The Open University, UK, (undated)
Clapp, J.A. et al	"Expert Systems for C^3I" Vol 1, USAF, Hanscom Air Force Base, Massachusetts, Oct 1985
Cromarty, A.S.	"What are current expert system tools missing?", CH2135-2/85/0000/0411 © 1985 IEEE
Greer, T.H.	"Artificial Intelligence: A New Dimension in EW", Defense Electronics, Sept, Oct 1985
van Lamsweerde A., and Dufor, P.	"Current Issues in Expert Systems" Academic Press, London, 1987 ISBN 0-12-714030-1
Marlin, C.	"Using Expert Systems in Crisis Management", Department of Computer Science, University of Adelaide, (undated)
Michie, D.	"Expert Systems", The Computer Journal, Vol 23, No 4, (undated)
Moroney, M.J.	"Facts from Figures", Penguin Books, 1962
Narain, S.	"MYCIN in a Logic Programming Environment", CH2017-2/84/0000/0192 © 1984 IEEE
Parsaye, K.	"The Next 700 Expert System Languages", CH2135-2/85/0000/0108 © 1985 IEEE
Tello, E.R.	"Guru", Byte, Aug 1986

APPENDIX A

ARRAYS AND STATISTICS

A.1 Requirement

A specific requirement for array processing arose from the use of XSHELL in the Intelligent Frequency Management System, where rules relating to each of many channels would otherwise require replication by that number, creating a significant maintenance problem. Other experience indicates that array processing will be a common requirement in expert systems, as it is in other software.

A.2 Description

Array subscripts are accommodated within the rule structure, using what would otherwise be term 2 of a numeric expression. This limits the structure of rule lines containing arrays.

The reader may have already gathered that a formal limited numeric expression structure was adopted in preference to one of free form, to avoid the need for a comprehensive expression interpreter. Such a capability was not considered necessary in an expert system, having logical rather than mathematical emphasis.

The form of a rule line in XSHELL is:

LH term [rel.op] [maths function] term1 [maths.op term2]

where the relational operator is required for an IF line only.

Either LH term or term1 can be an array in which case term2 is the array subscript. The subscript is displayed in brackets after the array as you would expect. The subscript may be a constant or a numeric fact. The subscript "each" implies a rule iteration through all members of the array. "Each" is represented internally by the constant 0.

The set of maths functions includes several array functions. If an array function is selected, then no subscript is needed for term1, and LH term may be an array.

These and other limitations need not be memorised, since XSHELL tightly controls what the user may select.

The several lines in the THEN part of a rule can be regarded as procedural, and so what is too complex for one line can be done in several, their order guaranteed.

A.2.1 Incomplete Arrays

XSHELL permits the assignment of values to particulai members of an array. It is therefore possible to leave some members undefined. XSHELL tests for the definition of an array at the end of the relevant rule search. Incompleteness is indicated by a screen diagnostic and execution continues with the whole array partly defined.

This is allowed so that arrays can be sized for the largest problem likely to be encountered by the domain.

A.2.2. Examples

These illustrate some of the array constructs allowed:

```
x = A(n)
x = LOG(A(n))
A9n) = y
A(n) = SIN(y)
A(each) = y
A(n) = MEAN(B)
x = STD DEVIATION(B) + y
```

where A and B are arrays, x is a numeric fact, and
Y and n are each a numeric fact or a constant.

APPENDIX B

FUNCTIONS

When you construct a rule containing a numeric expression you are given the option of a mathematical function. The array functions are in this list and XSHELL will limit your choice of operand according to your choice of function.

The maths and array functions are:

ABSVAL	returns the unsigned value
INTEGER	decimal portion is truncated
SIGN	returns 1, 0 or -1
SQROOT	square root
LOG	logarithm base 10
10↑	anti-log base 10
LN	natural log
e↑	natural anti-log
SIN	trig sine function
COS	cosine
TAN	tangent
ARCTAN	inverse tangent
INCREASED BY	LH term plus argument
DECREASED BY	LH term minus argument
MULTIPLIED BY	LH term times argument
DIVIDED BY	LH term divided by argument
SIZE	number of array members
SUM	sum of array members
INDEX OF MAX	subscript of maximum in array
INDEX OF MIN	subscript of minimum
MODE	most common value
RANGE`	maximum minus minimum
HARMONIC MEAN	)
MEAN DEVIATION	)
MEAN	)
STD ERR (MEAN)	)
MEDIAN	) for definitions see
STD ERR (MEDIAN)	) below.
STD DEVIATION	)
STD ERR (STD DEVIATION)	)
VARIANCE	)
STD ERR (VARIANCE)	)
COEFF/VARIATION	)
STD ERR(COEFF OF V)	)

Angles are specified in degrees

The harmonic mean is the number of array members (n) divided by the sum of the reciprocals of each member value. This is the average to use in cases like speeds over a given distance.

The mean deviation is the sum of the unsigned differences between each member and the mean, divided by n.

The mean is the arithmetic average; the sum of the members divided by n.

The median is the average of the maximum and the minimum.

The standard deviation (RMS deviation) s, is the square root of the average of the squares of the differences between each member and the mean.

The standard errors (STD ERR ...) are the standard deviation of the possible error in calculating the particular function from the samples in the array. In each case the error decreases as the square root of the array size.

Since the SIZE function is independent of the values of an array, a rule search for the array does not occur at this function.

DISTRIBUTION

	Copy No.
EXTERNAL	
Department of Defence	
Director General, Communications and Information Systems	1-2
Director, Communications and Information Systems (Army)	3
Defence Science and Technology Organisation	
Chief Defence Scientist)	4
Central Office Executive)	
Councellor, Defence Science, London	Cnt. Sht.
Counsellor, Defence Science, Washington	Cnt. Sht.
Scientific Adviser, Defence Central	Cnt. Sht.
Scientific Adviser, Defence Intelligence Organisation	Cnt. Sht.
Scientific Adviser - Army	Cnt. Sht.
Scientific Adviser - Air force	Cnt. Sht.
Scientific Adviser, Navy	Cnt. Sht.
INTERNAL	
Electronics Research Laboratory	
Director	5
Chief, Communications Division	6
Research Leader, Communications	7
Head, Switching Signalling and Networks	8
Chief Information Technology Division	9
Research Leader Information Systems	10
Technical Documentation Section, Communications Division	11-12
Libraries and Information Services	
Director of Publications, Australian Government Publishing Service	13
Librarian, Technical Reports Centre, Defence Central Library Campbell Park	14
Manager, Document Exchange Centre, Defence Information Services	15
Defense Technical Information Center, United States	16-27
Defence Research Information Centre, United Kingdom	28-29
Director Scientific Information Services, Canada	30
Ministry of Defence, New Zealand	31
National Library of Australia	32
Defence Science and Technology Organisation Salisbury, Main Library	33-34
Librarian Defence Signals Directorate Melbourne	35
British Library Document Supply Centre	36
Engineering Societies Library (USA)	37
Institution of Electrical Engineers (UK)	38
Spares	
DSTO Salisbury Main Library	39-44

Department of Defence

DOCUMENT CONTROL DATA SHEET

Page Classification
UNCLASSIFIED

Privacy Marking/Caveat

1a. AR Number AR-005-997	1b. Establishment Number ERL-0509-RE	2. Document Date MAY 90	3. Task Number -

4. Title

XSHELL - A GENERAL PURPOSE
EXPERT SYSTEM SHELL

5. Security Classification

Document	Title	Abstract
U	U	U

S (Secret) C (Confi) R (Rest) U (Unclass)

* For UNCLASSIFIED docs with a secondary distribution LIMITATION, use (L)

6. No. of Pages	26
7. No. of Refs.	-

8. Author(s)

Allan J. Mack

9. Downgrading/Delimiting Instructions

-

10a. Corporate Author and Address

Electronics Research Laboratory
PO Box 1600
SALISBURY SA 5108

10b. Task Sponsor

-

11. Officer/Position responsible for

Security........-........

Downgrading........-........

Approval for Release........-........

12. Secondary Release of this Document

APPROVED FOR PUBLIC RELEASE

Any enquiries outside stated limitations should be referred through DSTIC, Defence Information Services, Department of Defence, Anzac Park West, Canberra, ACT 2600.

13a. Deliberate Announcement

No Limitations

13b. Casual Announcement (for citation in other documents)

☒ No Limitation

□ Ref. by Author & Doc No only

14. DEFTEST Descriptors

Expert systems
Intelligent Frequency Management System

15. DISCAT Subject Codes

120901

16. Abstract

Available expert systems show some significant deficiencies when applied to practical applications requiring reliability and maintainability. This report describes an expert system, XSHELL, which addresses these problems and which has been successfully used in the Intelligent Frequency Management System.

Page Classification

UNCLASSIFIED

Privacy Marking/Caveat

16. Abstract (CONT.)

17. Imprint

Electronics Research Laboratory
PO Box 1600
SALISBURY SA 5108

18. Document Series and Number	19. Cost Code	20. Type of Report and Period Covered
REPORT 0509	603855	–

21. Computer Programs Used

None

22. Establishment File Reference(s)

None

23. Additional information (if required)