6-1-1989

An Expert System for Managing an Activated Sludge Wastewater Treatment Plant

Sandra C. Parker
University of Arkansas, Fayetteville

David G. Parker
University of Arkansas, Fayetteville

AN EXPERT SYSTEM FOR MANAGING AN ACTIVATED SLUDGE WASTEWATER TREATMENT PLANT

Sandra C. Parker
Industrial Engineering
and
David G. Parker
Civil Engineering
University of Arkansas
Fayetteville, AR 72701

Research Project Technical Completion Report

Project G-1549-05

The research on which this report is based was financed in part by the United States Department of the Interior as authorized by the Water Research and Development Act of 1984 (P.L. 98-242).

Arkansas Water Resources Research Center
University of Arkansas
113 Ozark Hall
Fayetteville, AR 72701

Publication No. 141

June, 1989

ABSTRACT

AN EXPERT SYSTEM FOR MANAGING AN ACTIVATED
SLUDGE WASTEWATER TREATMENT PLANT

A diagnostic expert system for an activated sludge wastewater treatment plant has been designed to link with a relational database management system for obtaining operational parameter values that are used by the program to diagnose operational problems that may occur in the process. The problems that are dealt with by the system are bulking sludge, floating sludge, defloculation, ashing, solids washout, foaming problems, high soluble effluent BOD and problems in the aeration system.

The link between the expert system and the database is accomplished via programming that is initiated by the expert system program. The operator of the system is not required to perform any action in order for the appropriate retrievals of operational parameter values to occur. The system is designed such that parameter values are retrieved from the database if such a database exists and contains appropriate values and, if no such database exists or if the appropriate values are not present, the operator is queried for the parameter values.

Since many wastewater treatment plants maintain database management systems for operational parameter values, such an expert system has advantages over stand alone systems. However, an override of the database query is possible, making the system useful for experimental queries and for training.

Sandra C. Parker and David G. Parker

Completion Report to the U.S. Department of the Interior, Reston, VA, June 1989.

Keywords -- Water Quality Control/ Water Quality Modeling/ Water Quality Monitoring/ Water Quality Standards

TABLE OF CONTENTS

INTRODUCTION

Expert systems are computer programs that solve problems by using expert knowledge. Such software has the ability to make judgements and respond to inquiries as a human would. An expert system must be able to make choices and respond to human communication. The common architecture of an expert system consists of a database, a knowledge base, and an interpreter or inference engine for interfacing the data and knowledge for making interpretations.

A. Purpose and Objectives

The purpose of the expert system described in this paper is to provide assistance to the personnel charged with the operation and control of an activated sludge treatment process. The most commonly used secondary wastewater treatment process is the activated sludge process. The activated sludge process is capable of producing a high quality effluent; however, proper operation and control is essential to achieve optimum performance. Studies have shown that most problems with activated sludge systems are due to faulty operation rather than faulty design (EPA-625/6-84-008 1982).

In the design of this expert system, the possible problems that may arise in an activated sludge treatment process have been taken into consideration. In the event that such a problem does arise, an operator may use the expert system to analyze the problem, generate information regarding the problem, and use this information to solve the problem and bring the activated sludge process back into control.

Some of the operational problems addressed in this expert system are bulking sludge, floating sludge, defloculation, ashing, solids washout, forming problems, high soluble effluent BOD, and aeration systems problems.

An operator of an activated sludge wastewater treatment process may use this system in the identification of the specific problem or problems that have arisen. The system will then determine the probable causes of the problems and suggest solutions for correcting the problems and bringing the process under control.

B. Related Research and Activities

Activated Sludge Analyzer (ASA)

A previous version of this expert system, called ASA, was reported earlier (Parker and Parker 1988). In the earlier version, now called ASA I, the system was designed to query the operator for all information concerning the operational parameters of the activated sludge treatment process. In this case, it would have been necessary for the operator to perform tests to obtain parameter values or to obtain such values from an existing database that may have been maintained by the treatment plant. Given that many wastewater treatment plants do, in fact, maintain process parameter data in a database management system, and given the widely recognized acceptance and use of relational database management systems, ASA has been redesigned and reprogrammed to access a relational database for operational parameter values if such a database of values is in place. The relational database management system that is used in the programming of the new expert system, called ASA II,is Rbase

5000 (Microrim 1985), however, the program could be adapted to access any relational database management system and some nonrelational systems as well.

METHODS AND PROCEDURES

ASA II

The new version of the activated sludge analyzer expert system, ASA II (Appendix), is designed to obtain some of the information that it requires to analyze problems by asking questions of the operator. This is descriptive information concerning the problem or problems that exist. Other information, however, is obtained from an operational database. This additional information consists of operational parameter values. The program is designed to query the database for necessary parameter values, accept any that are not over seven days old, and query the operator for any values that were not obtainable from the database. If no database exists, the system simply queries the operator for all of the requisite parameter values for analyzing the problem at hand.

ASA II is developed in Turbo Prolog (Borland 1986) with a link to Rbase that is partially accomplished via Turbo C routines. On execution, the Prolog program invokes Rbase by running a resident program called rbase.exe which takes instructions from a resident command file called rbase.dat. This program opens the wastewater treatment plant database, if one exists, retrieves any relevant parameter values, and creates a data file for retrieved data in which the data values are stored. Control is passed back to the Prolog program which then executes an appropriate routine in Turbo C (Borland 1988) which is designed to read the parameter values, reformat them in Prolog style, and output the reformatted parameter values into files from which the Prolog program may later access them if needed for a particular problem solution. This operation of retrieval, reformatting, and storage takes only a few seconds at the beginning of the Prolog program.

For the purposes of demonstrating the ability to link the Prolog expert system with an existing database, a test database was created using Rbase 5000. This database consists of several operational variables and the date of collection of the data. The variables that are included are dissolved oxygen (DO), pH, mixed liquor suspended solids (MLSS), and mean cell residence time (MCRT). Values for these parameters are retrieved from the database when the database does exist and when there are data values for the parameters that are not over seven days old. In the case that there are several values present within the last seven days, the program is written to chose the value associated with the most recent date.

PRINCIPAL FINDINGS AND SIGNIFICANCE

Advantages of Linking Expert System to Database

There are real advantages to a link between the expert diagnostic system and an existing database. Without the link, the operator would be required to obtain these parameter values either by querying the database or from existing hardcopy from the database. This activity would be time consuming, disruptive to the execution of the expert system, and would also introduce the potential for errors. The parameters could be either misread or entered incorrectly to the expert system. The expert system's ability to take data directly from an existing database is a distinct advantage over the necessity to manually input these data values.

The database that was created to use in this link is not designed as a database adequate to serve all of the data storage and reporting needs of a wastewater treatment plant. It is simply a small collection of variables, four operational parameters and date of collection, that was compiled as an example with which to demonstrate the link with the expert system. An actual wastewater treatment plant database management system design would need to be much more comprehensive than the experimental one used in this research.

The selection of data from the database that is no older than seven days is admittedly an arbritary decision. This selection criterion can be easily adjusted to any number of days that is desired. The statement that controls this criterion appears in the command file, rbase.dat, and can be adjusted to exercise a different criterion for age of data.

An added advantage of ASA II is that the operator has the option of overriding the database link if desired. This option could be useful in some situations. For example, one may desire to use the expert system to investigate 'what if' situations to develope a more thorough understanding of the activated sludge wastewater treatment process. Also, the system could be used in training new operators who may not have had adequate experience with this type of treatment process.

CONCLUSIONS

This research has resulted in the successful development of a link between an expert system and a relational database. The expert system is designed to monitor and diagnose problems in the operation of an activated sludge wastewater treatment plant. Given that many treatment plants maintain operational data in database management systems, an expert system that is designed to access such databases, when desired, has advantages over a stand alone diagnostic system. It is easier and more convenient for the operator to use, it can be used for experimental purposes as well as to solve real problems in the treatment process, and it can be used as an aid in training new operators who need to become more familiar with the process.

LITERATURE CITED

Borland. 1986. Turbo Prolog: The Natural Language of Artificial Intelligence, Borland International, Inc., Scotts Valley, CA.

Borland. 1988. Turbo C Reference Guide, Version 2.0, Borland International, Inc., Scotts Valley, CA.

Handbook for Improving POTW Performance Using the Composite Correction Program Approach. 1982. EPA-625/6-84-008, U.S. Envir. Protection Agency, Center for Envir. Research Information, Cincinnati, OH.

Microrim. 1985. RBASE Series 5000, User's Manual, Microrim, Inc., Bellevue, Washington.

Parker, David G. and Sandra C. Parker. 1988. ASA: An Expert System for Activated Sludge Analysis. In Microcomputer Knowledge-Based Expert Systems in Civil Engineering, edited by Hojjat Adeli, pp. 57-69. American Society of Civil Engineers.

APPENDIX

```
/*
     [1] THIS EXPERT SYSTEM CAN RETREIVE DATA FROM RBASE DATABASE
CALLED "WATER".  IF THE DATABASE DOES NOT EXIST THEN IT WILL
ASK THE REQUIRED VALUES FROM THE USER.
     IF THE DATABASE "WATER" EXISTS THEN THE FOLLOWING FILES
SHOULD BE PRESENT IN THE SAME DIRECTORY AS THIS EXPERT SYSTEM :-
1---> FILES FROM RBASE SYSTEM DISK 1.
2---> FILES FROM RBASE SYSTEM DISK 2.
3---> FILES FROM RBASE FILEGATEWAY DISK.
4---> RBASE.DAT FILE.
5---> DFORMAT(1,2,3,4).EXE FILES (THESE ARE THE FILES THAT FORMAT
      THE ASCII DATA OUTPUT BY WATER DATABASE SO THAT THIS EXPERT
      SYSTEM CAN READ IT.  THE SOURCE CODE FOR THESE PROGRAMS IS
      WRITTEN IN TURBO C).
*/
/*---------------------------------------------------------------------*/



code = 4800

domains
problem = symbol
reply = char
file = trialfile
data_term = do_(real); pH_(real); mlss_(real); mcrt_(real)



/*---------------------------------------------------------------------*/
/*
    [2] THERE ARE  TWO DATABASE PREDICATES THAT ARE USED IN THIS EXPERT
    SYSTEM.  positive PREDICATE WILL STORE IN DYNAMIC DATABASE  THE
    PROBLEM (SUCH AS do, mep, mlss, ETC) AND AN INTEGER CORRESPONDING
    TO ITS LEVEL.  rb PREDICATE WILL STORE IN DYNAMIC DATABASE THE
    VALUES RETREIVED FROM RBASE DATABASE IF ANY.
*/



database
positive(problem,integer)
rb(symbol,real)



/*---------------------------------------------------------------------*/
/*
     [3] THE FOLLOWING FILES FROM TURBO PROLOG TOOLBOX MUST BE PRESENT
     IN THE SAME DIRECTORY AS THIS EXPERT SYSTEM IF THE PROGRAM IS
     TO BE RUN FROM PROLOG ENVIRONMENT.  ONCE THE PROGRAM IS COMPILED
     IN AN EXECUTABLE FORM, THEN THESE FILES ARE NOT REQUIRED.
*/
```

```
include "tdoms.pro"
include "tpreds.pro"
include "menu.pro"


/*------------------------------------------------------------------------*/


predicates
go_list(integerlist)
idfy(integer)
check(problem,integer)
cause(problem,integer,integer)
ccause(problem,integer)
go
do(reply)
get_choice(real,integer,real,real)
forget1
forget2
rbase


/*------------------------------------------------------------------------*/


goal
go.


/*------------------------------------------------------------------------*/


clauses


/*------------------------------------------------------------------------*/


go :-
system("erase water*.dat"),    /* [4] ERASES ANY PREEXISTING WATER DATAFILES AND FAILS */
fail.


go :-
system("rbase -r"),    /* [5] GOES TO RBASE AND OUTPUTS APPROPRIATE DATA IF PRESENT AND FAILS AND FAILS */
fail.
```

```
go :-
system("dformat1"),    /* [6] FORMATS DATA FOR do_ PREDICATE SO THAT IT CAN BE READ BY PROLOG AND FAILS */
fail.


go :-
system("dformat2"),    /* [7] FORMATS DATA FOR ph_ PREDICATE SO THAT IT CAN BE READ BY PROLOG AND FAILS */
fail.


go :-
system("dformat3"),    /* [8] FORMATS DATA FOR mlss_ PREDICATE SO THAT IT CAN BE READ BY PROLOG AND FAILS */
fail.


go :-
system("dformat4"),    /* [9] FORMATS DATA FOR mcrt_ PREDICATE SO THAT IT CAN BE READ BY PROLOG AND FAILS */
fail.



go :- forget1.    /* [10] ERASES ALL THE PREEXISTING positive PREDICATES IN DYNAMIC DATABASE IF ANY AND FAILS.  SEE COMM
ENT [33]. */
go :- forget2.    /* [11] ERASES ALL THE PREEXISTING rb PREDICATES IN THE DYNAMIC DATABASE IF ANY AND FAILS.  SEE COMMEN
T [33]. */



go :-rbase.    /* [12] THIS RULE CHECKS IF THERE ARE ANY FORMATTED PREDICATES CONTAINING
                  VALUES OUTPUT BY RBASE "WATER" DATABASE.  IT FAILS IN THE END.  SEE COMMENT [34]. */




/*
     [13] FOLLOWING GO RULES ARE THE ONLY go RULES WHICH CAN SUCCEED. THESE MAKE
     THE WINDOW WITH A TITLE AND WILL PROMPT FOR A LIST OF SECONDARY CLARIFIER
     OBSERVATIONS. A CALL TO go_list PREDICATE WILL IDENTIFY PROBLEMS AND
     REMEDIES FOR EACH OF THE OBSERVATIONS.  FINALLY , THE USER WILL BE ASKED
     IF ANOTHER SESSION IS DESIRED.
*/



go :-
makewindow(1,32,66,"ASA : AN EXPERT SYSTEM FOR ACTIVATED SLUDGE ANALYSIS",0,0,25,80),
nl,nl,nl,write("                                          WELCOME"),nl,nl,nl,nl,nl,nl,
write("          PLEASE PRESS [RETURN] SO THAT WE CAN BEGIN CONSULTATION"),nl,nl,nl,
readdevice(keyboard),
readchar(_),
clearwindow,
write("                        SECONDARY CLARIFIER OBSERVATIONS"),nl,nl,
```

```
write("Use the up/down arrow keys to locate cursor."),nl,
write("Make your selection or selections using the return key."),nl,nl,
write("Press F10 key when all selections are completed."),nl,nl,nl,
write("1 - The sludge blanket in the clarifier is too high and/or"),nl,
write("    the sludge volume index (SVI) is over 150 ml/g."),nl,nl,
write("2 - Sludge and/or bubbles float to the surface of the clarifier"),nl,
write("    and/or a portion or all of the sludge rises to the surface"),nl,
write("    within four hours after the test is started."),nl,nl,
write("3 - A scum or ash floats on the surface of the clarifier."),nl,nl,
write("4 - The supernatent above the sludge blanket is cloudy and/or"),nl,
write("    settling test is cloudy  and/or contains poorly settling"),nl,
write("    suspended matter."),
menu_mult(25,66,7,7,["1","2","3","4"],"CHOICES",[],XXX),
go_list(XXX),
clearwindow,
write("Consultation over."),nl,nl,
write("Press [y] for another consultation."),nl,nl,
write("Press [n] to end the session."),nl,nl,
readchar(Reply),
do(Reply).

go :-
clearwindow,
write("Sorry the program is unable to analyze the problem from your responses."),nl,nl,nl,
write("Press [y] if you want another consultation."),nl,nl,nl,
write("Press [n] to end the session."),nl,nl,
readchar(Reply),
do(Reply).



/*------------------------------------------------------------------------*/
/*     [14] THESE RULES WILL LET THE EXPERT SYSTEM END THE SESSION
       OR TO HAVE ANOTHER SESSION DEPENDING UPON USER RESPONSE.
*/



do('n') :-
removewindow,nl,nl,nl,nl,nl,nl,nl,nl,nl,nl,nl,nl,nl,nl,nl,nl,nl,nl,nl,nl,nl,
write("Thanks for consulting me, Bye."),nl,nl,nl,
system("cls").

do('N') :-
removewindow,nl,nl,nl,nl,nl,nl,nl,nl,nl,nl,nl,nl,nl,nl,nl,nl,nl,nl,nl,nl,nl,
write("Thanks for consulting me, Bye."),nl,nl,nl,
system("cls").

do('y') :-
removewindow,
go.

do('Y') :-
removewindow,
```

```
go.

do(_) :-
clearwindow,
write("Please enter a valid choice as following :"),nl,nl,nl,
write("Press [y] if you want another consultation."),nl,nl,nl,
write("Press [n] to end the session."),nl,nl,
readchar(Reply),
do(Reply).




/*----------------------------------------------------------------------*/




go_list([]).          /* SEE COMMENT [13] */
go_list([H|T]) :-
idfy(H),
go_list(T).



/*----------------------------------------------------------------------*/
/*     [15] THIS RULE CALLS check RULE AND GETS A VALUE FOR VARIABLE
       Choice.  THIS VALUE OF Choice IS PASSED TO ccause(mef,Choice)
       TO DETERMINE WHAT ARE THE CAUSES AND SOLUTIONS TO mef PROBLEM.
*/



idfy(1):-
check(mef,Choice),
ccause(mef,Choice).

idfy(2)  :-        /* [16] SAME AS COMMENT [15] EXCEPT FOR mef REPLACED WITH nr */
check(nr,Choice),
ccause(nr,Choice).



/*     [17] THIS RULE IDENTIFIES THE SCUM OR ASH PROBLEM.  THE check RULES
       ARE CALLED TO GET THE VALUES OF C1, C2, C3, C4 WHICH ARE THEN USED
       TO CALL THE APPROPRIATE cause RULES TO FIND THE CAUSES AND SOLUTIONS
       TO THE SCUM OR ASH PROBLEM.
*/




idfy(3) :-
check(mep,C1),
check(do,C2),
```

```
check(mcrt,C3),
check(asa,C4),
clearwindow,
write("The problem seems to be SCUM or ASH on the surface."),nl,nl,
write("These are the causes and solutions :"),nl,nl,
cause(mep,C1,1),
cause(do,C2,1),
cause(mcrt,C3,5),
cause(asa,C4,5),
readchar(_),
clearwindow.



/*     [18] SIMILAR TO COMMENT [17].  CLOUDY EFFLUENT IS IDENTIFIED */



idfy(4) :-
check(mep,C1),
check(do,C2),
check(mcrt,C3),
clearwindow,
write("The problem seems to be CLOUDY EFFLUENT."),nl,nl,
write("These are the causes and solutions :"),nl,nl,
cause(mep,C1,1),
cause(do,C2,1),
cause(mcrt,C3,5),
readchar(_),
clearwindow.



/*----------------------------------------------------------------------*/
/*    [19] THIS RULE CHECKS IF THE PROBLEM HAS ALREADY BEEN IDENTIFIED
      AND INSERTED IN THE DYNAMIC DATABASE IN THE FORM OF positive
      PREDICATE.  SEE COMMENT [2].
*/



check(Prob,Z) :-
positive(Prob,Z),!.



/*    [20] THIS RULE WILL ASK THE CONDITION OF mef IF IT HAS NOT BEEN
      DETERMINED PREVIOUSLY.
*/



check(mef,Choice) :-
```

```
clearwindow,nl,
write("              MICROSCOPIC EXAMINATION FOR FILAMENTS"),nl,nl,nl,nl,
write("Pick appropriate condition"),nl,nl,
write("1 - FEW OR NO FILAMENTOUS ORGANISMS ARE PRESENT."),nl,nl,
write("2 - MANY OR EXCESSIVE FILAMENTOUS ORGANISMS ARE PRESENT."),nl,nl,
menu(25,10,7,7,["1","2"],"CHOICES",0,Choice),
asserta(positive(mef,Choice)).



/*
      [21] SIMILAR TO COMMENT [20] EXCEPT FOR mep.
*/



check(mep,Choice) :-
clearwindow,nl,
write("              MICROSCOPIC EXAMINATION FOR PROTOZOA"),nl,nl,nl,nl,
write("Pick appropriate condition"),nl,nl,
write("1 - PROTOZOA ARE PRESENT AND ACTIVE."),nl,nl,
write("2 - PROTOZOA ARE PRESENT BUT INACTIVE."),nl,nl,
write("3 - PROTOZOA ARE ABSENT."),nl,nl,
menu(25,10,7,7,["1","2","3"],"CHOICES",0,Choice),
asserta(positive(mep,Choice)).




/*      [22] THE NEXT THREE check RULES ARE ORDERED SO THAT IF do VALUE
        EXISTS IN THE DYNAMIC DATABASE (IN THE FORM OF rb PREDICATE)
        THEN THE USER WILL NOT BE ASKED TO INPUT IT (FIRST OF THE THREE
        check RULES).  IF THE do VALUE DOES NOT EXIST THEN THE USER WILL
        BE ASKED TO INPUT IT (SECOND check RULE).  UPON AN INVALID ENTRY
        THE USER IS GIVEN ANOTHER TRY (THIRD check RULE).
*/



check(do,Choice) :-
clearwindow,
rb(do,DO),
get_choice(DO,Choice,1.0,2.0),!,
write("THE DO LEVEL RETREIVED FROM DATABASE IS ",DO," mg/l"),nl,
readchar(_),
asserta(positive(do,Choice)).

check(do,Choice) :-
clearwindow,nl,
write("              WHAT IS DISSOLVED OXYGEN LEVEL IN AERATION TANK ?"),nl,nl,nl,nl,
write("              Enter DO level in mg/l : "),
readreal(DO),
get_choice(DO,Choice,1.0,2.0),!,
asserta(positive(do,Choice)).
```

```
check(do,Choice) :-
write("              ENTER DISSOLVED OXYGEN LEVEL IN A CORRECT FORMAT"),nl,
write("              PLEASE HIT [RETURN] FOR ANOTHER TRY..."),
readchar(_),
check(do,Choice).




/*
     [23] SIMILAR TO COMMENT [22].  DOES IT FOR mlss.
*/



check(mlss,Choice) :-
clearwindow,
rb(mlss,MLSS),
write("THE MLSS LEVEL RETREIVED FROM DATABASE IS ",MLSS," mg/l."),nl,
readchar(_),
get_choice(MLSS,Choice,1000.0,4000.0),!,
asserta(positive(mlss,Choice)).

check(mlss,Choice) :-
clearwindow,nl,
write("              WHAT IS THE LEVEL OF MIXED LIQUOR SUSPENDED SOLIDS ?"),nl,nl,nl,nl,
write("              Enter MLSS level in mg/l. : "),
readreal(MLSS),
get_choice(MLSS,Choice,1000.0,4000.0),!,
asserta(positive(mlss,Choice)).

check(mlss,Choice) :-
write("              ENTER THE LEVEL OF MIXED LIQUOR SUSPENDED SOLIDS"),nl,
write("              IN A CORRECT FORMAT."),nl,
write("              PLEASE HIT [RETURN] FOR ANOTHER TRY..."),
readchar(_),
check(mlss,Choice).




/*
    [24] SIMILAR TO COMMENT [22].  WORKS FOR mcrt.
*/



check(mcrt,Choice) :-
clearwindow,
rb(mcrt,MCRT),
write("THE MCRT RETREIVED FROM DATABASE IS ",MCRT," days."),nl,
readchar(_),
get_choice(MCRT,Choice,2.0,15.0),!,
asserta(positive(mcrt,Choice)).
```

```
check(mcrt,Choice) :-
clearwindow,nl,
write("              WHAT IS MEAN CELL RESIDENCE TIME ?"),nl,nl,nl,nl,
write("              Enter MCRT in days : "),
readreal(MCRT),
get_choice(MCRT,Choice,2.0,15.0),!,
asserta(positive(mcrt,Choice)).

check(mcrt,Choice) :-
write("              ENTER THE MEAN CELL RECIDENCE TIME IN A CORRECT FORMAT"),nl,
write("              PLEASE HIT [RETURN] FOR ANOTHER TRY..."),
readchar(_),
check(mcrt,Choice).




/*
     [25] SIMILAR TO COMMENT [22].  WORKS FOR pH.
*/




check(pH,Choice) :-
clearwindow,
rb(pH,PH),
write("THE pH LEVEL RETREIVED FROM IS ",PH),nl,
readchar(_),
get_choice(PH,Choice,6.5,7.5),!,
asserta(positive(pH,Choice)).

check(pH,Choice) :-
clearwindow,nl,
write("              WHAT IS THE LEVEL OF pH IN THE AERATION TANK"),nl,nl,nl,nl,
write("              Enter pH : "),
readreal(PH),
get_choice(PH,Choice,6.5,7.5),!,
asserta(positive(pH,Choice)).

check(pH,Choice) :-
write("              ENTER THE LEVEL OF pH IN A CORRECT FORMAT"),nl,
write("              PLEASE HIT [RETURN] FOR ANOTHER TRY..."),
readchar(_),
check(pH,Choice).




/*
     [26] SIMILAR TO COMMENT [20].  DOES IT FOR bod.
*/
```

```
check(bod,Choice) :-
clearwindow,nl,
write("             BOD/NITROGEN/PHOSPHOROUS RATIO IN AERATION TANK"),nl,nl,nl,nl,
write("Pick appropriate condition"),nl,nl,
write("1 - BOD/N < 100/5 AND BOD/P < 100/1"),nl,nl,
write("2 - BOD/N > 100/5"),nl,nl,
write("3 - BOD/P > 100/1"),nl,nl,
menu(25,10,7,7,["1","2","3"],"CHOICES",0,Choice),
asserta(positive(bod,Choice)).



/*
    [27] SIMILAR TO COMMENT [20].  DOES IT FOR nnisc.
*/



check(nnisc,Choice) :-
clearwindow,nl,
write("             NITRATE NITROGEN IN INFLUENT TO SECONDARY CLASSIFIER"),nl,nl,nl,nl,
write("Pick appropriate condition"),nl,nl,
write("1 - NITRATE PRESENT."),nl,nl,
write("2 - NITRATE ABSENT."),nl,nl,nl,
menu(25,10,7,7,["1","2"],"CHOICES",0,Choice),
asserta(positive(nnisc,Choice)).



/*
    [28] SIMILAR TO COMMENT [20].  WORKS FOR npat.
*/



check(npat,Choice) :-
clearwindow,nl,
write("             NITRATE PRODUCTION IN AERATION TANK"),nl,nl,nl,nl,
write("Pick appropriate condition"),nl,nl,
write("1 - NITRATE CONCENTRATION INCREASES BETWEEN INFLUENT"),nl,
write("    AND EFFLUENT IN AERATION TANK."),nl,nl,
write("2 - NITRATE CONCENTRATION DOES NOT INCREASE BETWEEN"),nl,
write("    INFLUENT AND EFFLUENT IN THE AERATION TANK."),nl,nl,nl,
menu(25,10,7,7,["1","2"],"CHOICES",0,Choice),
asserta(positive(npat,Choice)).



/*
    [29] SIMILAR TO COMMENT [20]. WORKS FOR nr.
*/
```

```
check(nr,Choice) :-
clearwindow,nl,
write("             NITRIFICATION REQUIREMENT"),nl,nl,nl,nl,
write("Pick appropriate condition"),nl,nl,
write("1 - NITRIFICATION IS REQUIRED AND/OR DESIRED."),nl,nl,
write("2 - NITRIFICATION IS NOT REQUIRED AND NOT NECESSARILY DESIRED"),nl,
menu(25,10,7,7,["1","2"],"CHOICES",0,Choice),
asserta(positive(nr,Choice)).




/*
     [30] SIMILAR TO COMMENT [20]. DOES IT FOR asa.
*/



check(asa,Choice) :-
clearwindow,nl,
write("             APPEARANCE OF SCUM OR ASH"),nl,nl,nl,nl,
write("Pick appropriate condition"),nl,nl,
write("1 - GREASE CONCENTRATION IN INFLUENT IS LESS THAN 100 mg/l."),nl,nl,
write("2 - GREASE CONCENTRATION ININFLUENT IS GREATER THAN 100 mg/l."),nl,nl,
menu(25,10,7,7,["1","2"],"CHOICES",0,Choice),
asserta(positive(asa,Choice)).



/*----------------------------------------------------------------------*/



ccause(mef,1) :-      /* SEE COMMENT [15] */
clearwindow,
check(mep,C1),
check(do,C2),
check(mlss,C3),
check(mcrt,C4),
check(pH,C5),
clearwindow,
write("The problem seems to be NON-FILAMENTOUS BULKING SLUDGE."),nl,nl,
write("These are the causes and solutions :"),nl,nl,
cause(mep,C1,1),
cause(do,C2,1),
cause(mlss,C3,1),
cause(mcrt,C4,1),
cause(pH,C5,1),
readchar(_),
clearwindow.

ccause(mef,2) :-         /* SEE COMMENT [15] */
clearwindow,
check(do,C1),
check(mlss,C2),
```

```
check(mcrt,C3),
check(pH,C4),
check(bod,C5),
clearwindow,
write("The problem seems to be FILAMENTOUS BULKING SLUDGE."),nl,nl,
write("These are the causes and solutions :"),nl,nl,
cause(do,C1,1),
cause(mlss,C2,2),
cause(mcrt,C3,1),
cause(pH,C4,2),
cause(bod,C5,2),
readchar(_),
clearwindow.

ccause(nr,1) :-          /* SEE COMMENT [16] */
clearwindow,
check(do,C1),
clearwindow,
write("The problem seems to be FLOATING SLUDGE."),nl,nl,
write("These are the causes and solutions :"),nl,nl,
write("Cause is probably denitrification in the clarifier."),nl,
write("If possible, increase recirculation rate to reduce"),nl,
write("the time that sludge remains in the clarifier."),nl,nl,
cause(do,C1,3),
readchar(_),
clearwindow.

ccause(nr,2) :-          /* SEE COMMENT [16] */
check(nnisc,C2),
C2 = 1,
check(npat,C3),
clearwindow,
cause(npat,C3,4),
readchar(_),
clearwindow.

ccause(nr,2) :-     /* SEE COMMENT [16] */
clearwindow,
write("Sorry, the program is unable to analyze the problem"),nl,nl,
write("according to your responses or data retreived."),nl,nl,
write("Please hit return to proceed."),nl,nl,
readchar(_).


/*----------------------------------------------------------------------*/
/*
     [31] THE cause RULES HAVE THREE ARGUMENTS. THE FIRST ARGUMENT
     INDICATES WHICH PROBLEM'S SOLUTION IS GIVEN BY THE CAUSE RULE.
     THE SECOND ARGUMENT CORRESPONDS TO THE USER RESPONSE TO ONE
     OF THE check RULES OR THE Choice VALUE FROM get_choice RULE.
     THE THIRD ARGUMENT RELATES EACH cause RULE TO ONE OR MORE OF
     THE idfy, ccause, OR cause RULES FROM WHICH cause RULES ARE
     CALLED.
*/
```

```
cause(mep,1,1).

cause(mep,2,1) :-
write("Possible cause is resent toxic shock."),nl,nl,
write("Check for toxic conditions."),nl,nl.

cause(mep,3,1) :-
write("Possible cause is toxic conditions."),nl,nl,
write("Check for toxic conditions."),nl,nl.

cause(do,1,1) :-
write("Possible cause low DO."),nl,nl,
write("Increase DO to between 1 and 2 mg/l."),nl,nl.

cause(do,2,1).

cause(do,3,1).

cause(mlss,1,1) :-
write("Possible cause low MLSS."),nl,nl,
write("Increase MLSS to between 1000 and 4000 mg/l."),nl,nl.

cause(mlss,2,1).

cause(mlss,3,1) :-
write("MLSS not cause but should probably lower MLSS."),nl,nl.

cause(mcrt,1,1) :-
write("Possible cause high MCRT."),nl,nl,
write("Lower MCRT to between 2 and 15 days."),nl,nl.

cause(mcrt,2,1).

cause(mcrt,3,1).

cause(pH,1,1) :-
write("Possible cause low pH."),nl,nl,
write("Increase pH to between 6.5 and 7.5"),nl,nl.

cause(pH,2,1).

cause(pH,3,1) :-
write("Possible cause high pH."),nl,nl,
write("Lower pH to between 6.5 and 7.5"),nl,nl.

cause(mlss,1,2) :-
write("MLSS not cause but should probably increase MLSS to between 1000 and 4000 mg/l."),nl,nl.

cause(mlss,2,2).

cause(mlss,3,2) :-
write("Possible cause high MLSS."),nl,nl,
write("Should lower MLSS to between 1000 and 4000 mg/l."),nl,nl.
```

```
cause(pH,1,2) :-
write("Possible cause low pH."),nl,nl,
write("Increase pH to between 6.5 and 7.5."),nl,nl.

cause(pH,2,2).

cause(pH,3,2) :-
write("pH not cause but should probably lower pH to between 6.5 and 7.5"),nl,nl.

cause(bod,1,2).

cause(bod,2,2) :-
write("Possible cause low nitrogen."),nl,nl,
write("Should add nitrogen to influent."),nl,nl.

cause(bod,3,2) :-
write("Possible cause low phosphorus."),nl,nl,
write("Should add phosphorus to influent."),nl,nl.

cause(do,1,3) :-
write("Increasing DO to above 2 mg/l may help"),nl,
write("to reduce denitrification in clarifier."),nl,nl.

cause(do,2,3) :-
write("Increasing DO to above 2 mg/l may help to reduce "),nl,
write("denitrification in the clarifier."),nl,nl,
write("Decreasing DO below 1 mg/l may reduce or stop nitrification."),nl,nl.

cause(do,3,3) :-
write("Decreasing DO to 1 mg/l may reduce or stop nitrification."),nl,nl.

cause(npat,1,4) :-
check(mcrt,C5),
check(do,C6),
clearwindow,
write("The problem seems to be FLOATING SLUDGE."),nl,nl,
write("These are the causes and solutions :"),nl,nl,
write("The cause is probably denitrification in the secondary"),nl,
write("clarifier."),nl,nl,
write("If possible, increase recirculation rate to"),nl,
write("reduce the time that sludge remains in the clarifier."),nl,nl,
cause(mcrt,C5,4),
cause(do,C6,4).


cause(mcrt,1,4).

cause(mcrt,2,4) :-
write("Reduce MCRT to stop nitrification if possible."),nl,nl.

cause(mcrt,3,4) :-
cause(mcrt,2,4).

cause(do,1,4) :-
```

```
write("Increasing DO to 2 mg/l may help"),nl,
write("to reduce denitrification in clarifier."),nl,nl.

cause(do,2,4) :-
write("Increasing DO to 2 mg/l may help to reduce "),nl,
write("denitrification in the clarifier."),nl,nl,
write("Decreasing DO below 1 mg/l may stop nitrification."),nl,nl.

cause(do,3,4) :-
write("Decreasing DO to 1 mg/l may stop nitrification."),nl,nl.

cause(npat,2,4) :-
check(do,C7),
clearwindow,
write("The problem seems to be FLOATING SLUDGE."),nl,nl,
write("These are the causes and solutions :"),nl,nl,
write("The cause is probably denitrification in the secondary clarifier."),nl,nl,
write("If possible, increase recirculation rate to"),nl,
write("reduce the time that sludge remains in the clarifier."),nl,nl,
write("Nitrate in the influent is causing problem."),nl,nl,
cause(do,C7,4).

cause(mcrt,1,5) :-
write("MCRT is too low."),nl,nl,
write("Increase MCRT to between 2 and 15 days."),nl,nl.

cause(mcrt,2,5).

cause(mcrt,3,5) :-
write("MCRT is too high."),nl,nl,
write("Lower MCRT to between 2 and 15 days."),nl,nl.

cause(asa,1,5).

cause(asa,2,5) :-
write("Possible cause is high grease content in the influent."),nl,nl,
write("Reduce grease concentration in influent to less than 100 mg/l."),nl,nl.



/*---------------------------------------------------------------------------*/
/*
     [32] get_choice RULE TAKES FOUR ARGUMENTS.  IF THE LEVEL
     IS LESS THAN LOWER (LIMIT) THEN IT RETURNS Choice = 1.
     IF LEVEL IS GREATER THAN UPPER (LIMIT) THEN Choice = 2.
     IF THE LEVEL LIES BETWEEN THE TWO LIMITS THEN Choice = 3.
*/



get_choice(Level,Choice,Lower,_) :-
Level < Lower,
Choice = 1.
get_choice(Level,Choice,_,Upper) :-
Level > Upper,
```

```
Choice = 3.
get_choice(Level,Choice,Lower,Upper) :-
Level >= Lower,
Level <= Upper,
Choice = 2.



/*-------------------------------------------------------------------------*/
/*
      [33] SEE COMMENTS [10] AND [11].

*/



forget1 :-
retract(positive(_,_)),forget1.
forget2 :-
retract(rb(_,_)),forget2.



/*-------------------------------------------------------------------------*/
/*
      [34] SEE COMMENT [12].
*/



rbase :-
closefile(trialfile),
openread(trialfile,"water13.dat"),
readdevice(trialfile),
readterm(data_term,do_(A1)),
closefile(trialfile),
asserta(rb(do,A1)),
fail.
rbase :-
closefile(trialfile),
openread(trialfile,"water23.dat"),
readdevice(trialfile),
readterm(data_term,ph_(A2)),
closefile(trialfile),
asserta(rb(pH,A2)),
fail.
rbase :-
closefile(trialfile),
openread(trialfile,"water33.dat"),
readdevice(trialfile),
readterm(data_term,mlss_(A3)),
closefile(trialfile),
asserta(rb(mlss,A3)),
fail.
rbase :-
```

```
closefile(trialfile),
openread(trialfile,"water43.dat"),
readdevice(trialfile),
readterm(data_term,mcrt_(A4)),
closefile(trialfile),
asserta(rb(mcrt,A4)),
fail.


/*-----------------------------------------------------------------------*/
```