

UNIVERSITY OF CALIFORNIA
Los Angeles

**Development of a
Knowledge-Based Expert System for
Process Control of Nitrification in the
Activated Sludge Process**

**A dissertation submitted in partial satisfaction of the
requirements for the degree Doctor of Philosophy
in Civil Engineering**

by

Naci H. Ozgur

1991

© Copyright by

Naci H. Ozgur

1991

The dissertation of Naci H. Ozgur is approved.




D. Okrent



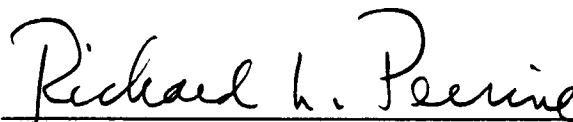
M. Elimelech



W. Karplus



M. Wurtele



R. L. Perrine



M. K. Stenstrom, Committee Chair

University of California, Los Angeles
1991

TO
A VERY SPECIAL PERSON
IN MY LIFE,
MARGARET JONES-RYAN

Table of Contents

List of Figures	viii
List of Tables	xii
List of Symbols	xiv
Acknowledgements	xviii
Vita	xix
Abstract	xx
1 Introduction	1
2 Literature Review	4
2.1 Activated Sludge	5
2.1.1 Activated Sludge Process Description	6
2.1.1.1 Nature of Refinery Wastewater Influent	8
2.1.1.2 Design Considerations	13
2.1.1.3 Aeration Tanks	14
2.1.1.4 Mean Cell Residence Time	16
2.1.1.5 Solids in Aeration Tanks	17
2.1.1.6 Sludge Volume Index	18
2.1.1.7 Food to Microorganisms Ratio	19
2.1.1.8 Dissolved Oxygen Concentration	20
2.1.1.9 Organic Loading	21
2.1.1.10 Biological View	21
2.1.1.11 Nutrient Requirements	23
2.1.1.12 Oxygen Requirements	24
2.1.1.13 Common Microorganisms	24
2.1.1.14 Filamentous Microorganisms	25
2.1.1.15 Advanced Options: Mixing, Flexibility	26
2.1.2 Nitrification Process Description	27
2.1.2.1 Nitrification Microbiology	28
2.1.2.2 Nitrification Stoichiometry and Kinetics	37
2.1.2.3 Nitrification in the Activated Sludge	48

2.1.3	Problems with Activated Sludge Process	65
2.1.3.1	Process Monitoring	66
2.1.3.2	Poor Operations	68
2.1.3.3	Nitrification Failure	72
2.1.3.4	Bulking	76
2.1.3.5	Other Settlement Problems	80
2.1.3.6	Foaming	82
2.1.3.7	Minor Causes of Activated Sludge Problems	84
2.1.4	Process Control and Troubleshooting	87
2.1.4.1	Air Control	87
2.1.4.2	Sludge Return	90
2.1.4.3	Sludge Wasting	96
2.1.4.4	RAS Chlorination	100
2.1.4.5	Nutrient Addition	102
2.1.4.6	Polymer Addition	103
2.1.4.7	Advanced Process Control Options	104
2.1.5	Mathematical Models for Activated Sludge	106
2.2	Expert Systems	112
2.2.1	Introduction to Expert Systems	114
2.2.2	Structure of Expert Systems	119
2.2.2.1	Expert System Languages	120
2.2.2.2	Hardware Requirements	121
2.2.2.3	Expert System Shells	123
2.2.2.4	Manipulation of Logic	128
2.2.2.5	Knowledge Acquisition	135
2.2.2.6	Knowledge Base Search Techniques	139
2.2.2.7	Uncertainty in the Rules	153
2.2.2.8	Knowledge Base Development and Organization	162
2.2.2.9	Knowledge Base Testing	168
2.2.2.10	Data Base Structure	171
2.2.3	Statistical Models: Bayesian	174
2.2.4	Mathematical Models in Expert Systems	186
2.2.5	Expert Systems Applications	190
2.2.5.1	Expert Systems in Water Quality Management	190
2.2.5.2	Expert Systems in Other Fields	195

2.2.6	Selected Shell: Personal Consultant™ Plus	198
3	Development of the Activated Sludge Process eXpert (ASPX)	202
3.1	Building the Foundation of ASPX	204
3.1.1	Knowledge Acquisition Technique for ASPX	206
3.1.2	Structure of ASPX	209
3.1.3	ASPX Knowledge Base	212
3.1.3.1	Rules in the ASPX Knowledge Base	216
3.1.3.2	Handling Uncertainty in ASPX	218
3.1.4	ASPX Graphical User-Interface	222
3.1.5	ASPX Data Base	226
3.1.5.1	ASPX Statistical Model	232
3.1.5.2	ASPX Mathematical Model	239
3.1.6	ASPX Expert System	252
3.1.7	ASPX Report Generation Module	260
3.2	Testing the ASPX Knowledge Base	264
3.2.1	Initial Testing	265
3.2.2	Implementation and Intermediate Testing	267
3.2.3	Advanced Testing	268
4	Application of ASPX to the Selected Refinery Plant	270
4.1	Description of the Selected Plant	274
4.2	Process Control of the Plant Prior to ASPX	280
4.3	Plant Process Control by ASPX	285
4.3.1	Ammonia Control by ASPX	292
4.3.2	Sludge Age Control by ASPX	297
4.3.3	Dissolved Oxygen Control by ASPX	303
4.3.4	pH Control by ASPX	309
4.3.5	Phosphate Control by ASPX	312
4.3.6	Temperature Control by ASPX	316
4.4	Future uses of ASPX in the selected plant	319

5 Conclusions	326
References	330
Appendix A: ASPX knowledge base logic trees	361
Appendix B: Result codes	372
Appendix C: Questionnaire results	401
Appendix D: Determination of nitrification state value	416
Appendix E: ASPX knowledge base rules	421

List of Figures

Figure 2-1: Basic activated sludge process diagram	7
Figure 2-2: Nitrogen conversion by microorganisms	30
Figure 2-3: First and zero order reactions in substrate-limited growth kinetics	42
Figure 2-4: Monitoring of the activated sludge process	67
Figure 2-5: Career ladder for wastewater operators in the State of California	70
Figure 2-6: (a)Normal condition; (b)No filaments present; (c)Filament bridging (Type I bulking); (d)Presents of few flocs (Type II bulking)	78
Figure 2-7: Control of dissolved oxygen concentration in activated sludge process	89
Figure 2-8: Control of the return activated sludge rate	91
Figure 2-9: Control of the waste activated sludge rate	97
Figure 2-10: Chemical addition control scheme	105
Figure 2-11: Basic components of an expert system	116
Figure 2-12: Requirements relevant to choosing a shell	127
Figure 2-13: (a) Knowledge representation by semantic networks; (b) Example of semantic networks	129
Figure 2-14: Knowledge representation by frame-based systems	133
Figure 2-15: The loss of knowledge during knowledge acquisition	137
Figure 2-16: Knowledge base component within the expert system	164
Figure 2-17: Frame structure of knowledge base	166

Figure 2-18: Data base component within expert system	172
Figure 2-19: Data base functions during expert system consultation . . .	175
Figure 2-20: Definition of <i>State</i> and <i>Sample</i> variables	178
Figure 2-21: Use of mathematical models in expert systems	187
Figure 2-22: Example of nomenclature used in the logic tree creation . .	200
Figure 3-1: The frame structure of the ASPX knowledge base	208
Figure 3-2: Interaction between the different modules of the ASPX program	210
Figure 3-3: The flowchart of the ASPX consultation	211
Figure 3-4: ASPX sub-frames and the number of rules and suggestions for each sub-frame	213
Figure 3-5: A sample screen from the ASPX program showing the three main menu options	225
Figure 3-6: Example frequency analysis for the dissolved oxygen with the high, low and normal values	235
Figure 3-7: Schematic diagram of the reactor setup for the model developed by Poduska and Andrews (1975)	241
Figure 3-8: A sample mathematical model simulation result to predict the effluent ammonia concentration for the next 3 days	248
Figure 3-9: Determination of the minimum MCRT value as it is calculated by the mathematical model	251
Figure 3-10: An example of determining the nitrification state	255
Figure 3-11: The dialog box displayed by ASPX to run the expert system shell with different options	258

Figure 3-12: An example turnover sheet generated by ASPX	261
Figure 3-13: A sample graphical data base record: Influent ammonia concentration for the last 24 days with the upper and lower 5% and the median values	262
Figure 4-1: The process diagram of the selected refinery treatment plant	275
Figure 4-2: The process control scheme of the ASPX program	289
Figure 4-3: Legends used for the logic trees	293
Figure 4-4: The effluent ammonia concentration process control logic tree	295
Figure 4-5(a): The sludge age process control logic tree, part 1	299
Figure 4-5(b): The sludge age process control logic tree, part 2	300
Figure 4-6(a): The dissolved oxygen process control logic tree, part 1 . .	304
Figure 4-6(b): The dissolved oxygen process control logic tree, part 2 .	305
Figure 4-7: The logic tree for the process control of pH level	310
Figure 4-8: The effluent phosphate process control logic tree	314
Figure 4-9: The logic tree for the process control of temperature	318
Figure A-1: Complete logic tree for ammonia control	362
Figure A-2(a): Complete logic tree for the control of sludge age, part 1	363
Figure A-2(b): Complete logic tree for the control of sludge age, part 2	364
Figure A-2(c): Complete logic tree for the control of sludge age, part 3	365
Figure A-3(a): Complete logic tree for the control of DO, part 1	366

Figure A-3(b): Complete logic tree for the control of DO, part 2	367
Figure A-3(c): Complete logic tree for the control of DO, part 3	368
Figure A-4: Complete logic tree for the control of pH concentration . . .	369
Figure A-5: Complete logic tree for the control of phosphate addition . .	370
Figure A-6: Complete logic tree for the control of temperature	371

List of Tables

Table 2-1: Some of the activated sludge parameters found in the refinery treatment plant influent (compiled from Dyer and Mignone, 1983; and EPA, 1980)	10
Table 2-2: Some US EPA priority pollutants that are found in the refinery treatment plant influent (compiled from Vernick <i>et al.</i> , 1984; and Dyer and Mignone, 1983)	12
Table 2-3: Typical design parameters for activated sludge process	15
Table 2-4: Optimum condition requirements for the microorganisms in the activated sludge systems for organic compound removal and for nitrification	23
Table 2-5: Some genera of nitrifying bacteria (Adapted from Stainer <i>et al.</i> , 1986)	34
Table 2-6: Some of micronutrients required for nitrifying bacteria	36
Table 2-7: Some of the chemicals which are found in the refinery treatment plant influent, their inhibitory concentrations, and inhibitory effects on the nitrifying bacteria	38
Table 2-8: Kinetic coefficients reported in the literature for <i>Nitrosomonas</i>	45
Table 2-9: Kinetic coefficients reported in the literature for <i>Nitrobacter</i>	46
Table 2-10(a): Typical range of kinetic coefficient values for <i>Nitrosomonas</i> and <i>Nitrobacter</i> under various temperatures in the pure cultures (pH= 7.0-8.0)	47
Table 2-10(b): Typical range of kinetic coefficient values for <i>Nitrosomonas</i> and <i>Nitrobacter</i> under various temperatures in the activated sludge (pH= 6.4-8.5)	47
Table 2-11: The estimation of nitrifier fraction for given BOD ₅ /TKN ratios (EPA, 1975)	52

Table 2-12: Effect of temperature on the nitrification kinetic parameters and the reaction rates	58
Table 2-13: Range of $K_{O,A}$ and b_A reported in the literature	60
Table 2-14: Activated sludge process sampling locations, analysis and frequencies (Adapted from WPCF, 1980)	69
Table 2-15: The probable causes of filamentous bulking are suggested as (compiled from WPCF, 1987 and Richard, 1989):	80
Table 2-16: Some of the expert system shells available for personal computers (partly compiled from Wigan, 1986)	125
Table 2-17: Questions relevant to choosing forward and backward chaining (Adapted from Pedersen, 1989)	152
Table 2-18: Example of the Bayesian statistical table	183
Table 2-19: Expert systems in water related field	192
Table 2-20: Expert systems in other fields	196
Table 3-1: Error and the default CFs for the input parameters in ASPX .	219
Table 3-2: ASPX data base parameters: Type of measurements, the number of locations, frequency of measurement, and whether the parameter is required or optional	228
Table 3-3: Example of the Bayesian statistical table for determining the minimum MCRT value	238
Table 3-4: Values used for the constants in Eq. 3-9 and 3-10	246
Table 3-5: Parameters that have limiting values in the ASPX program and the default values chosen for the selected refinery plant	253
Table 4-1: The list of process control parameters, process control equipment, location and the monitoring equipment for the ASPX process control technique	287

List of Symbols

- $\{H_3PO_4\}_x$ = Required amount of phosphoric acid at time x , gpm
 a, b, c = Parameters
 Air flowrate $_x$ = Air flowrate needed to reach DO_x , SCFM
 b_A = Endogenous decay coefficient, 1/d
 $BOD_{5,x}$ = BOD_5 concentration at time x , mg/L
 C = Constant return ratio, %
 C_∞^* = Saturation concentration of oxygen, mg/L
 Caustic $_x$ = Required caustic rate for Q_x , gpm
 $C_{F/M}$ = Constant F/M ratio, 1/d
 CF = Confidence factor
 premise CF = CF of the *IF* block
 conclusion CF = CF of the *THEN* block
 CF (rule) = Rule's combined CF
 C_{MCRT} = Constant MCRT value, d
 $C_{NaOH,x}$ = Concentration of caustic solution at time x , %
 C_{vss} = Constant MLVSS concentration, mg/L
 DO_x = Dissolved oxygen concentration at time x , mg/L
 f = Fraction of nitrifiers
 F/M = Food to microorganisms ratio, 1/d
 Influent pH $_x$ = Influent pH level at time x
 K_i = Inhibition constant, mg/L
 K_{L,a_x} = Volumetric mass-transfer coefficient for DO_x , 1/h
 K_{NH} = Half-saturation constant for NH_4^+ , mg/L
 K_{NH}^* = Temperature adjusted half-saturation constant for
 NH_4^+ , mg/L
 K_O = Half-saturation concentration for dissolved oxygen, mg/L
 $K_{O,A}$ = Dissolved oxygen saturation constant for autotrophic
 K_s = Half-saturation constant, mg/L
 Level $_{c,x}$ = Level of caustic solution in tank at time x , ft
 Level $_{p,x}$ = Level of phosphoric acid in tank at time x , ft
 m = Parameter
 MCRT = Sludge age, days
 μ = Specific growth rate, 1/d

- μ_A = Specific growth rate of autotrophs (*Nitrosomonas*), 1/d
 μ'_A = Net specific growth rate for autotrophs, 1/d
 μ''_A = Temperature adjusted specific growth rate, 1/d
 μ'''_A = pH corrected specific growth rate of autotrophs, 1/d
 μ_m = Maximum specific growth rate, 1/d
 $\mu_{m,A}$ = Maximum specific growth rate of autotrophs, 1/d
 n = Total number of events observed in the past
 $\text{NH}_4^+_{,in}$ = Ammonia concentration in the influent, mg/L
 $\text{NH}_4^+_{,out}$ = Ammonia concentration in the effluent, mg/L
 NR = Nitrification rate, mg $\text{NH}_4^+ \text{-N}$ /mg MLVSS · d
 P = Efficiency of solid-liquid separation
 $P [\text{Sample}] =$ is prior probability of *Sample* regardless of whether *State_j* has occurred or not (prior probability)
 $P [\text{State}_j | \text{Sample}] =$ is conditional probability of having *State_j* when *Sample* exists (posterior probability)
 $P [\text{Sample} | \text{State}_i] =$ is conditional probability of *Sample* for specific *State_i*, (all the states which are observed in the past, $i = 1, 2, \dots, m$)
 $P [\text{State}_i] =$ is prior probability of *State_i*, (all the states which are observed in the past) regardless of whether *Sample* exists (prior probability)
 $P [\text{Sample} | \text{State}_j] =$ is conditional probability of observing *Sample* when *State_j* has occurred (conditional probability)
 $P [\text{State}_j] =$ is prior probability of *State_j*, regardless of whether *Sample* exists or not (prior probability)
pH = pH of the wastewater
 $\text{PO}_4\text{-P}_x$ = Required phosphate at time x
 P_{vss} = Percent VSS content (MLVSS/MLSS), %
 Q = Flowrate, gpm
 Q_{KLax} = Site-specific coefficient for $K_L a_x$
 Q_o = Influent flowrate, gpd
 $Q_{p,min}$ = Minimum flowrate capacity of the sludge pumps, gpm
 Q_r = Return activated sludge flowrate, gpm
 Q_w = Waste sludge flowrate, gpm
 Q_x = Flowrate at time x , gpm

- R = Gas constant, cal/mole · °K
 R_{20} = Reaction rate at temperature 20°C
 R_T = Reaction rate at temperature T°C
 r_A = Rate of autotrophic bacteria growth,
 mg *Nitrosomonas*/L · d
 r_{\max} = Nitrogen utilization rate, mg N/mg TSS · h
 r_N = NH_4^+ -N utilization rate, mg NH_4^+ -N/L · d
 r_c = Radius of caustic storage tank, ft
 r_p = Radius of phosphoric acid storage tank, ft
 S = Limiting substrate concentration, mg/L
 $S_{\text{BOD},e}$ = BOD₅ concentration in the effluent, mg/L
 $S_{\text{BOD},o}$ = BOD₅ concentration in the influent, mg/L
 $S_{\text{BOD},o}/S_{\text{NH},o}$ = BOD₅/TKN ratio
 S_{NH} = Ammonia concentration, mg NH_4^+ -N/L
 $S_{\text{NH},e}$ = NH_4^+ -N concentration in the effluent, mg/L
 $S_{\text{NH},o}$ = NH_4^+ -N concentration in the influent, mg/L
 S_o = Influent BOD₅ concentration, mg/L
 S_o = Concentration of dissolved oxygen, mg/L
 T = Temperature of the wastewater, °K
 T' = Temperature in °C
 t_{ON} = Duration of waste sludge pump operations, hrs
 θ_H = Hydraulic retention time, hrs
 TOC_x = TOC concentration at time x, mg/L
 V = Volume of aeration basin, gal
 $V_{\text{H}_3\text{PO}_4}$ = Volume of phosphoric acid delivered during time₁-time₂,
 gal
 V_{NaOH} = Volume of caustic delivered during time₁-time₂, gal
 x = Number of times *Condition A* is observed in the past
 X = Mixed liquor suspended solids, mg/L
 X_A = Concentration of autotrophic bacteria, mg *Nitrosomonas*/L
 X_e = Effluent TSS concentration, mg/L
 X_H = Heterotroph concentration, mg/L
 $X_{\text{H,VSS}}$ = VSS fraction of the heterotrophic biomass, mg/L
 X'_{VSS} = Target MLVSS concentration, mg/L
 X_{VSS} = Aeration tank MLVSS concentration, mg/L

X_w = Waste sludge suspended solids, mg/L

Y_A = Maximum yield coefficient for autotrophs,
mg autotrophic biomass produced/mg NH_4^+ -N consumed

Y_H = Maximum yield coefficient for heterotrophs

ACKNOWLEDGEMENTS

Without the love and guidance of my parents, Professor Sabiha Özgür and Professor Tuğrul Özgür I would never have made it this far. Thank you for instilling in me the strength to pursue my goals and the confidence to succeed. Sizleri çok seviyorum ve bana çalışma zevkini tattırdığınız için teşekkürler. I could not possibly express my full appreciation enough to a very special person in my life, Margaret Jones-Ryan. Without her help I could not have completed this dissertation. Thank you for being there for me when I needed you the most. I also extended my deepest appreciation to Dr. Maureen Jones-Ryan for the many hours she spend fixing my typos and correcting my grammar. You did a superb job!

To my advisor Professor Michael K. Stenstrom who was very helpful through my many revisions and supported me during the course of my studies, a very special thank you. Many thanks to Miriam and Debby for easing the bureaucratic red tape of the doctoral process. I also would like to thank my advisor in Turkey, Dr. Orhan Uslu for stimulating my interest in the environmental engineering field. Finally, I wish good luck to Weber who will continue to develop the expert system program where I left off.

This research was funded by the BP America graduate fellowship program and the NSF-funded Hazardous Substance Control Engineering Research Center.

VITA

- November 14, 1963 Born, Izmir, Turkey
- 1984 B.S., Environmental Engineering
Dokuz Eylul University
Izmir, Turkey
- 1984 Valedictorian of graduating class
Dokuz Eylul University
Izmir, Turkey
- 1987-1990 Teaching Assistant
University of California
Los Angeles, California
- 1987-1990 Research Assistant
University of California
Los Angeles, California
- 1988 M.S., Engineering
University of California
Los Angeles, California

PUBLICATIONS

- Neethling, J.B., Ozgur, N. H. and Almog, I.J. (1988). Cost Estimates for Removing Radon from Water Using Granular Activated Carbon. Report to the State of California, Department of Health Services.
- Neethling, J.B., Ozgur, N. H. and Almog, I.J. (1988). Cost Estimates for Removing Bentazon from Water Using Granular Activated Carbon. Report to the State of California, Department of Health Services.
- Neethling, J.B., Ozgur, N. H. and Almog, I.J. (1988). Cost Estimates for Removing Freon 11 from Water Using Granular Activated Carbon. Report to the State of California, Department of Health Services.

ABSTRACT OF THE DISSERTATION

**Development of a
Knowledge-Based Expert System for
Process Control of Nitrification in the
Activated Sludge Process**

by

Naci H. Ozgur

**Doctor of Philosophy in Civil Engineering
University of California, Los Angeles, 1991
Professor Michael K. Stenstrom, Chair**

Knowledge-based system technology has recently been introduced in activated sludge plant operations. This dissertation develops a knowledge-based expert system entitled *Activated Sludge Process eXpert* (ASPX) for the process control of nitrification in a selected petroleum refinery activated sludge plant. This knowledge-based system has been developed exclusively to diagnose and correct problems with the nitrification process using rule-based knowledge, statistics, and a mathematical model.

The knowledge base consists of frames for each of the six chosen process control parameters. For each parameter the process control strategy is executed by the knowledge compiled in the logic tree form. The data base

functions such as the data entry forms, the statistical evaluation of the data base records, and the mathematical model simulation of the effluent ammonia concentration are also included. The user-interface enables operators who are not familiar with computers to use all the functions that the program offers.

Testing is performed in three steps which include initial, intermediate, and advanced testing procedures. Initial testing is performed by the developer based on the suggestions given by the program for several hypothetical problem cases. Intermediate and advanced testing is performed after the implementation of the computer program to the selected refinery plant. Testing is performed on-site using the real-time data base values and real-time observations.

Development of the knowledge-based expert system in this dissertation brings us one step closer to solving activated sludge problems. Innovative applications of expert system technology provide state-of-the-art computer solutions to wastewater treatment plant operations. Twenty-years or more of practical experience is made available to inexperienced operators through a knowledge-based expert system.

1 Introduction

This dissertation develops a knowledge-based expert system exclusively for the selected refinery treatment plant for the diagnosis and correction of nitrification process problems. The selected refinery treatment plant is typical in that it suffers from a periodic loss of nitrification. Operator rotation in the plant compounds the problem by eliminating the proper practice of a steady process control technique. Corrective measures are needed to replace the plant's current process control technique and improve the overall process quality.

The use of artificial intelligence in controlling and diagnosing problems with wastewater plants is a recent development in both the computer and wastewater engineering fields. The use of a computer which is capable of making decisions comparable to the most experienced operators in a plant is a newly applied process control technique.

Past experience with activated sludge plants questions the use of the activated sludge process with the existing process control strategies. Surveys conducted by the General Accounting Office (1980) showed that 87% of the wastewater treatment plants included in the survey were not in compliance with their National Pollutant Discharge Elimination System (NPDES) permits. Another study conducted by Junkins *et al.* (1983) showed that approximately 50% of the municipal wastewater treatment plants violated their NPDES permits for various controlled pollutants. This same study suggested that most of the problems are associated with poor operations and inadequate process control techniques.

Compliance with the standards in refinery treatment plants is more difficult since the refinery process-water may contain higher concentration of toxic or inhibitory chemicals, high oil and grease content, and high concentrations of other pollutants. Any violation of an NPDES permit may cause significant damage to the environment. Furthermore, public awareness of violations has motivated the refinery management to explore new process control technologies to reduce the risk of violations. Government agencies have recently put additional strain on the refinery management by assigning more complex and strict requirements for the discharge water quality as well as issuing large fines in case of violations. The collective impact is to provide additional incentives to comply with NPDES permits.

Improved process control of the operations of the activated sludge process are one way of meeting more restrictive NPDES permits. One state-of-the-art solution is the use of expert system technology to improve process control. In this dissertation, a knowledge-based expert system is developed for the process control of the nitrification process in the selected refinery process-water treatment plant.

Expert system technology is a relatively new concept in the wastewater treatment plant field. The first attempt of using the expert system concept in the wastewater field was seen in the late-1970s (Beck *et al.*, 1978) but it took about a decade for the first real-time applications to be developed (Geselbracht *et al.*, 1988; Berthouex *et al.*, 1989; Patry and Chapman, 1989). This dissertation adapts the newly applied expert system

technology as a process control technique for the selected activated sludge process.

This dissertation will first introduce the activated sludge process operations including plant design, process control techniques, and problems with the operations. Second, the literature review will present the historical development of the expert system technology, expert system program structure and functions, and possible application fields. Third, a knowledge-based system will be developed by creating rules for the process control of nitrification, and by developing the statistical and mathematical models for the data base parameters. The user-interface will also be presented, including available functions of the user-interface and the innovative presentation of the expert system capabilities. Finally, the testing and the implementation procedures will be discussed including the on-site testing and possible real-time applications.

2 Literature Review

Expert systems technology was born in the early-1950s primarily to imitate "human thinking." Since then, duplication of human brain functions has been the chief goal of many *Artificial Intelligence* (AI) experts. Pioneers in the AI field discovered the potential of applying human-like reasoning to computers (McCorduck, 1979). Later, many researchers established different approaches to create a machine that "thinks" (Barr and Feigenbaum, 1981; Siddall, 1984; Harmon and King, 1985).

Expert system technology has been newly applied to wastewater treatment plants. The first application of using expert knowledge on modeling and operations of wastewater treatment plants is seen in the late-1970s (Beck *et al.*, 1978). It took approximately 10 years for the first real-time application to be successfully performed (Geseibracht *et al.*, 1988; Patry and Chapman, 1989). Many others have applied expert systems to different areas in wastewater treatment: operation control strategies with expert systems (Gall and Patry, 1988); application of the statistical models within expert systems (Berthouex *et al.*, 1989); designing wastewater treatment plants (Rossman, 1989); and diagnosing activated sludge problems (Johnston, 1985; Maeda, 1985; Jenkins and Jowitt, 1987; Chan and Koe, 1991). The work of several authors established a new path for the best control strategy of systems in both water resources and environmental engineering applications (Reboh *et al.*, 1982; Barnwell *et al.*, 1986; Mikroudis *et al.*, 1986; Rossman and Siller, 1987).

Sludge bulking, perhaps the most common problem in activated sludge process, has received the most attention in the literature. Recent works focused greatly on diagnosis and possible control of the sludge bulking problem (Gall and Patry, 1988; Chan and Koe, 1991). Correlation between sludge bulking and treatment plant design was investigated by Geselbracht *et al.* (1988). The authors employed a *rule-based* expert system to find the likelihood of having sludge bulking in various wastewater treatment plant designs. The expert systems approach has the potential to resolve those problems most frequently encountered today by wastewater treatment operators. Application of expert systems is a state-of-the-art solution for the process control of the activated sludge process (Horan and Eccles, 1986).

In the literature, both expert systems and the activated sludge process have been studied separately and extensively. This literature review will first present background information on activated sludge, and the nitrification process. Expert system technology will next be discussed in detail including structural properties, development aspects, and functions of expert systems.

2.1 Activated Sludge

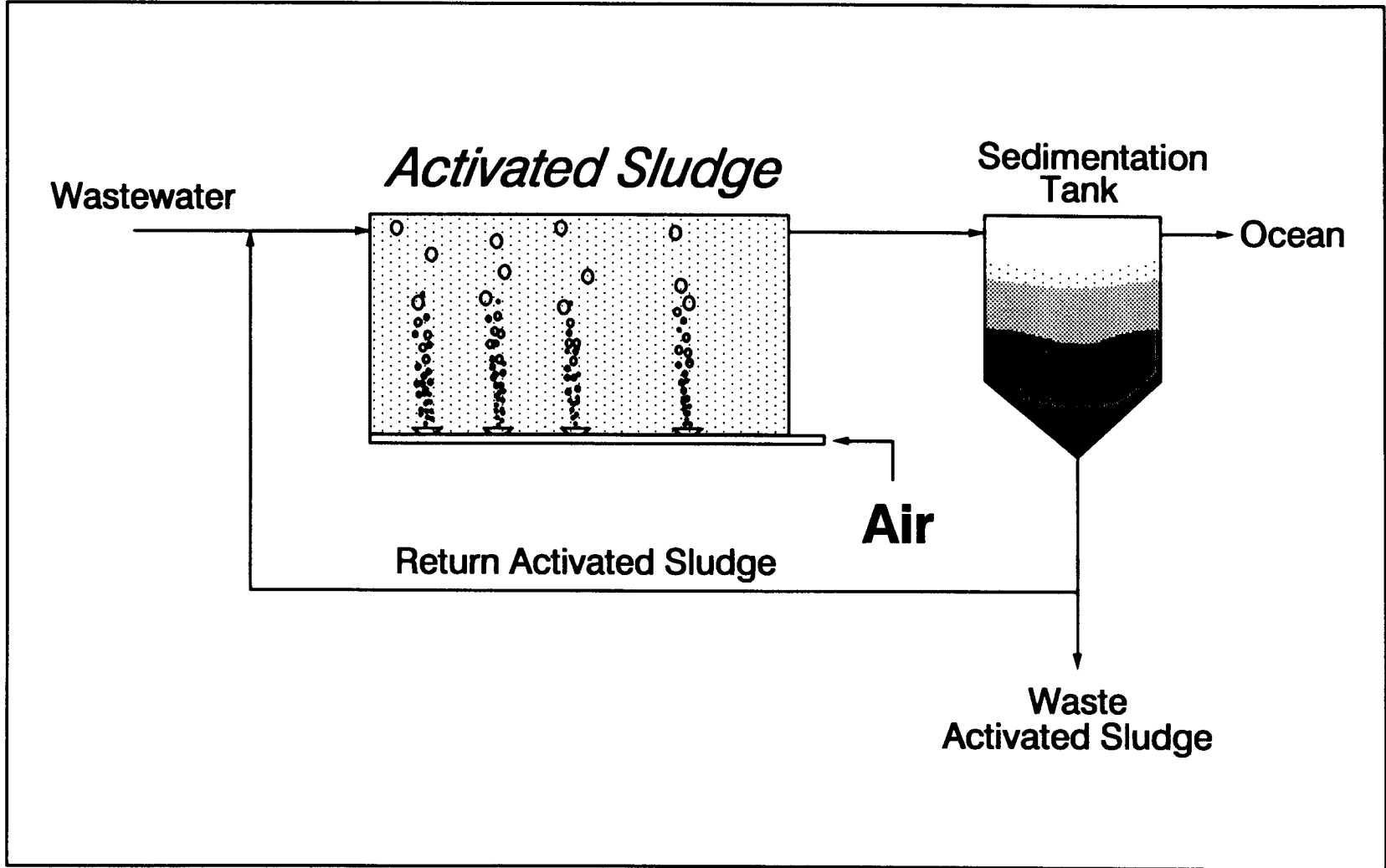
The literature review on the activated sludge process is presented here. The activated sludge process parameters, the problems with activated sludge processes, the process control and troubleshooting procedures, and

the mathematical models for the activated sludge process are discussed in the following sections.

2.1.1 Activated Sludge Process Description

Activated sludge is one type of aerobic biological system procedure and the most frequently used procedure to clean wastewater today. To achieve a clean effluent from activated sludge treatment plants, the process should be capable of at least two major functions which are as follows: (1) stabilize the organic compounds by oxidation or conversion to biomass; (2) remove inorganic compounds such as ammonia and phosphorus or convert them to less harmful forms.

The activated sludge procedure was developed in the early-1900s. The word "activated" refers to the activation of the sludge (*i.e.*, bacteria) by the addition of oxygen to the water. The liquid (*i.e.*, wastewater and bacteria) in the activated sludge tanks is referred to as "mixed-liquor." There have been several modifications to the activated sludge process. The most common ones are conventional, step aeration, and continuous-flow stirred-tank reactor (Metcalf and Eddy, 1991). In this dissertation, we will focus primarily on the conventional activated sludge process. The process consists of an aerated tank and a sedimentation tank. The basic process diagram is shown in Figure 2-1.



7

Figure 2-1: Basic activated sludge process diagram

The activated sludge tanks are aerated by sub-surface or surface aerator systems. There are several types of aerators, all of which try to supply adequate dissolved oxygen to the water for the bacteria to thrive. The wastewater flows through the tank while the bacteria is consuming the unwanted organic matter from the wastewater. The liquid is then transferred to sedimentation tanks where the bacteria settles to the bottom. The clean water portion is discharged to an ocean or sometimes it is further treated by advanced treatment processes.

2.1.1.1 Nature of Refinery Wastewater Influent

Wastewater is generated by two major sources: municipal and industrial. Municipal wastewater is generated by domestic water use and contains mostly dissolved biodegradable organics. Industrial wastewater may contain complex organics, toxic chemicals, inorganic substances, and chemicals which are used by the source factory during the production process.

This dissertation will focus on the refinery wastewater treatment to understand the significance of the wastewater characteristics on the nitrification process. Incoming wastewater to the treatment plant has certain identifiable characteristics including; *flowrate (Q)*, *Total Organic Carbon (TOC)*, *Biological Oxygen Demand (BOD)*, *Chemical Oxygen Demand (COD)*, *pH*, *Total Suspended Solids (TSS)* (APHA *et al.*, 1989).

All the above mentioned parameters are related to each other, however, their relationship changes from plant to plant depending on the unit process type and geographical location. Because great fluctuations are observed in almost all the characteristics of the wastewater, the plant operator has difficulty in maintaining proper operations. Moreover, frequent changes in the influent characteristics make the treatment process' behavior difficult to understand. Consequently, the operators must make decisions based on an ever-changing range of variables.

Furthermore, refinery plant operators face more difficult tasks than the municipal plant operators because, first, the refinery influent flowrate may fluctuate more than the municipal wastewater and, second, the contaminants in the refinery process-water have more potential to create operational problems than the contaminants found in the municipal wastewater. Finally, the refinery treatment plants are more likely to receive a toxic shock which destroys the living bacteria population, consequently, by creating a substantial effect on the plant performance.

Table 2-1 shows some of the activated sludge process' parameters, the expected range of concentrations, and the typical values for each parameter. The concentrations in Table 2-1 reflect the condition of the refinery process-water after the API separators (see section 4.1). The parameters shown in Table 2-1 are widely used in the activated sludge process control and the process performance evaluation.

Table 2-1: Some of the activated sludge parameters found in the refinery treatment plant influent (compiled from Dyer and Mignone, 1983; and EPA, 1980)

Activated sludge parameters	Concentration (mg/L)	
	Range	Typical*
Total Organic Carbon	25 - 240	51.1
BOD ₅	10 - 556	114
Chemical Oxygen Demand	75 - 3,330	261
pH (units)	6 - 11	---
Total Suspended Solids	6 - 490	---
Nitrogen, as ammonia	1 - 168	14.5
Oil and grease	3 - 775	44.1
Sulfides	0.1 - 32	1.24

*Median values for the refineries with the integrated process

The refinery waste process-water contains high levels of organic compounds which are measured by TOC, BOD, and the COD tests. When a high organic load is received by the activated sludge process, it is difficult to handle incoming load because, in most of the cases, the maximum plant capacity is exceeded. Complete COD reduction cannot be achieved under the current operational conditions. Consequently, excessive COD escapes to the plant's effluent water.

The pH is one of the important parameters in the refinery process-water treatment plants. The pH control is necessary because the refinery produces high pH water which causes disturbances in the bacteria growth.

Fluctuations of incoming pH generate additional difficulty to maintain pH within the desired range. Consequently, frequent adjustment to pH controlling equipment may be necessary in the refinery process-water treatment plants.

Refinery process-water may contain significant amount of ammonia due to pH control chemicals introduced in the oil refining processes. High ammonia concentrations are eliminated by the nitrification process which will be discussed later in the text in detail. Although refinery process-water contains excessive amount of nitrogen, it is usually low on other nutrients such as phosphate, which is essential for the bacteria growth. In most cases phosphate has to be added using phosphoric acid or trisodium phosphate.

The refinery treatment plant may also receive considerable amount of oil and grease in the influent water. The oil separators help prevent high oil content from entering the aeration basins. If the oil escapes into the treatment unit processes, the mechanical equipment failure and the problems with the biomass are more frequently observed.

During the refining of crude oil, toxic chemicals may be introduced to the process-water. The refinement process either uses water which comes in contact with the chemicals or the chemicals escape to the process-water by an accidental leakage and end up in the treatment plant. If the treatment plant receives toxic chemicals in substantial concentrations, the chance of having problems with the activated sludge process greatly increase.

Table 2-2: Some US EPA priority pollutants that are found in the refinery treatment plant influent (compiled from Vernick *et al.*, 1984; and Dyer and Mignone, 1983)

Some of the compounds found in the refinery treatment plant influent	Concentration ($\mu\text{g/L}$)	
	Range	Typical
Phenol	UD* - 200,000	1,500
Toluene	10 - 15,000	1,300
Chromium	1 - 2,000	700
Benzene	100 - 5,800	430
Zinc	UD - 3,600	360
Acenaphthene	37 - 3,000	270
Phenanthrene	30 - 660	170
Cyanide	UD - 5,250	150
Lead	10 - 4,900	140
Ethylbenzene	28 - 9,000	130
Arsenic	5 - 1,200	120
2,4-Dimethylphenol	16 - 9,300	80
Copper	4 - 2,275	65
Nickel	1 - 700	20
Chrysene	0.1 - 20	15
Fluoranthene	3 - 9	7

*Undetectable concentration

Analysis of the content of the refinery process-water reveals an extensive list of organic compounds and metals which may be toxic to the bacteria in the refinery treatment plant. The term toxic chemical is addressed here as the chemicals which have toxic or inhibitory effect on the activated sludge biomass.

Table 2-2 shows some of the chemicals in the US Environmental Protection Agency (EPA) priority pollutant list that are found in the refinery process-water treatment plant influent, the expected range of concentrations, and the typical concentration of each chemical. Table 2-2 lists the chemicals found in the refinery treatment plant influent in the order of the highest concentration to the lowest concentration. Some of these chemicals have an inhibitory effect on the nitrifying bacteria. The inhibitory effects are discussed in section 2.1.2.1.

2.1.1.2 Design Considerations

Designing the plant is a critical step in to completing an optimally reliable system. Any error made in the design of the plant has a direct influence on the plant's performance (WPCF, 1977; Chambers, 1982; Pitman, 1991). The design should allow several different modification possibilities for the process, if the original design does not produce the desired results. The operator should be able to switch tanks, divide flow, and control in/out structure for a more efficient process control. Overall, the design should be *flexible*.

The second most important property of the design is ease of control. Any operator should be able to learn to control the system within a reasonable time period. Operations should be as easy as possible so that if any problems occur, the corrective action can be taken immediately.

There are several computer programs in the literature which can assist a plant designer make a wise choice (Eiler and Smith, 1978). The design parameters are extensively researched and well documented in the literature. Table 2-3 shows basic design parameters and values for the conventional activated sludge process design. More details on the parameters can be found in the literature (Schroeder, 1977; WPCF, 1977; Metcalf and Eddy, 1991; Qasim, 1985; Tchobanoglous and Schroeder, 1985; WPCF, 1985).

The parameters in Table 2-3 will be briefly discussed, so that the importance of each parameter can be understood.

2.1.1.3 Aeration Tanks

Aeration tanks are usually designed uncovered, and open to the atmosphere. Air is supplied as dissolved oxygen to the bacteria by two primary methods: mechanical aerators or by diffusers. The mechanical aerators, such as surface aerators and brush aerators, irrigate the surface of the water mechanically. The concentration of O₂ in water can be controlled by adjusting the speed of the rotors.

Table 2-3: Typical design parameters for activated sludge process

Process Components or Variables	Section Number	Typical Values from Literature	Reference
Aeration Basin Depth (ft) Width (ft)	2.1.1.3	15 - 25 23 - 38	Metcalf and Eddy, 1991
Mean Cell Retention Time (d)	2.1.1.4	5 - 15	Metcalf and Eddy, 1991
MLSS (mg/L)	2.1.1.5	1,500-4,000	WPCF, 1977
SVI (mg/L)	2.1.1.6	40 - 150	WPCF, 1990
F/M (lb BOD ₅ /lb MLVSS · d)	2.1.1.7	0.2 - 0.4	Qasim, 1985
RAS Rate (% of Influent flowrate)	2.1.1.7	30 - 100	WPCF, 1977
Organic Loading (lb BOD ₅ /1000 ft ³ · d) [†]	2.1.1.9	20 - 60	WPCF, 1977
Oxygen Required (lb/lb BOD ₅ removed)	2.1.1.12	1.4 - 1.6	WPCF, 1977

[†] @ 3,000 mg/L MLSS

Another type of aeration is executed by diffused air aerators. This type of aeration is performed by forcing air or O₂ through the water body in the form of bubbles. The bubbles are formed by either diffusers or jet aerators. The degree of diffusion can be controlled by the amount of air fed through the diffusers.

Aeration is required to supply O₂ to the bacteria and to mix the liquor in the basin. If there are any "dead spots" in the basin, there will be anaerobic/anoxic conditions in the poorly mixed areas. This will cause an undesired odor and may also cause sludge bulking. A strategy for proper mixing is important to achieving a clean effluent. Design geometry and layout of the tanks also affect the performance of the procedure.

2.1.1.4 Mean Cell Residence Time

The most important design parameter is the *mean cell residence time* (MCRT) also interchangeably stated as *sludge age* (θ_c) or *solids retention time* (SRT). The MCRT is the ratio between the total solids in the system and the total solids leaving the system. The MCRT can be defined in an equation form as shown below:

$$MCRT = \frac{\text{Total Solids in Aeration Basin}}{\left(\frac{\text{Total Solids Leaving with Effluent} + \text{Total Solids in Waste Stream}}{\text{Total Solids in Waste Stream}} \right) \cdot \text{day}} \quad (2-1)$$

$$MCRT = \frac{V X}{(Q X_e + Q_w X_w) \times 24 \times 60} \quad (2-2)$$

where MCRT = Sludge age, days

V = Volume of aeration basin, gal

X = Mixed liquor suspended solids, mg/L

Q = Flowrate, gpm

X_e = Effluent TSS concentration, mg/L

Q_w = Waste sludge flowrate, gpm

X_w = Waste sludge suspended solids, mg/L

The MCRT is frequently used to operate the activated sludge plants. One of the most widely used process control technique is to maintain a constant MCRT in the system. For optimum results an average MCRT should be selected (see Table 2-3). This type of process control technique, however, requires 3 to 5 MCRT for changes to take place.

The MCRT can be controlled by adjusting the waste sludge rate (see Figure 2-1). Many treatment plants today follow the MCRT process control method because it is easy to execute and it is one of the most effective controlling strategy for the activated sludge process.

2.1.1.5 Solids in Aeration Tanks

The most important solids in the wastewater field are: suspended and the volatile fraction of the suspended solids. The total suspended solids in the aeration tank are named *mixed-liquor suspended solids* (MLSS). It refers to the amount of solids in a certain volume of the water (mg of solids/L).

The second most important parameter is *mixed-liquor volatile suspended solids* (MLVSS). The volatile fraction of the suspended solids is assumed to be the organic part (*i.e.*, cell mass) of the total solids. Therefore, MLVSS is frequently used for measuring the bacteria concentration. The

MLVSS is approximately equal to 75-85% of the MLSS concentration (Qasim, 1985).

2.1.1.6 Sludge Volume Index

The *sludge volume index* (SVI) is used to describe the settling characteristics of the sludge. The SVI test is often used as a process control parameter to determine the RAS rate.

The SVI is expressed in a unit of volume occupied by a gram of sludge (mL/g). The sludge settlement would normally yield a clear cut separation between the water and the sludge blanket. However, if the sludge has any problems, such as bulking, pinpoint floc, or ashing, this interface between the sludge blanket and the water may not be seen clearly.

Although the SVI test has been used widely and frequently in the treatment plants, the test has some problems. First, the SVI reading is taken only once at the 30 minute mark. One value does not characterize the complex sludge settlement process efficiently. Secondly, the use of glass or plastic graduated cylinders introduces a percentage of error into the testing procedure. Error is initiated by the cohesion forces between the narrow container's walls and the solid particles in the settling sludge.

2.1.1.7 Food to Microorganisms Ratio

The process control parameter, called *food to microorganisms ratio* (F/M), helps the operator control the system efficiently. The F/M ratio is expressed as the amount of organic food (lb BOD₅) available for the amount of microorganisms present (lb MLVSS) per day. The F/M ratio can be formulated by the following equation:

$$F/M = \frac{\text{Biodegradable substrate}}{\text{Microorganisms present in the aeration tanks}} \quad (2-3)$$

$$F/M = \frac{(Q) (S_o) (24) (60)}{V X_{vss}} \quad (2-4)$$

where F/M = Food to microorganisms ratio, 1/d

S_o = Influent BOD₅ concentration, mg/L

X_{vss} = MLVSS concentration, mg/L

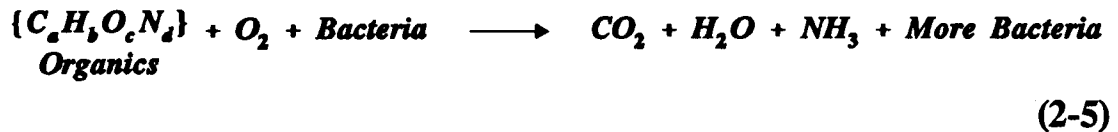
One of the most common process control strategies is to keep a constant F/M ratio in the aeration basins (WPCF, 1990). Once the optimum F/M ratio is known (see Table 2-3), the operator maintains that ratio at all times. Since the operator does not have control of the incoming BOD loading, MLVSS is adjusted for a given BOD₅ concentration.

The MLVSS in the basin can be controlled by adjusting the rate of Return Activated Sludge (RAS) rate (see Figure 2-1) to maintain a desired F/M ratio. Depending on the organic loading and the present MLVSS in the basin, the RAS rate can be as high as 100% of the influent flowrate (see

Table 2-3). Because of the great fluctuations of the incoming BOD₅ concentration, it is difficult to keep the F/M ratio constant.

2.1.1.8 Dissolved Oxygen Concentration

Bacteria in the aeration tank needs oxygen to consume the organic substrate and to thrive. The basic reaction for the consumption of organics by the bacteria can be formulated as:



As a result of the above equation, the organics are consumed by the bacterial respiration (i.e., oxidation) and new bacterial cells are produced by the assimilation process (i.e., synthesis). The oxygen is supplied through the diffusers at the bottom of the wastewater aeration tanks in bubble form. Based on the two-film mass transfer theory, the oxygen diffuses from the gas phase to the liquid phase (Danckwerts, 1970). The concentration of the dissolved oxygen in the liquid phase is directly proportional to the concentration of the oxygen in the gas phase.

The dissolved oxygen concentration can be controlled by either adjusting the speed of the air pump or throttling the air pipes. Although the use of pure oxygen systems (i.e., higher oxygen concentration in the gas phase)

will efficiently supply the dissolved oxygen concentration in aeration tanks (Stenstrom *et al.*, 1989), the air pump systems are more widely used to aerate wastewater because of their lower operational and maintenance costs.

2.1.1.9 Organic Loading

The amount of organic matter present in wastewater is commonly measured by the 5-day Biological Oxygen Demand (BOD₅) test, the Chemical Oxygen Demand (COD) test or, more conveniently, by the Total Organic Carbon (TOC) test (APHA *et al.*, 1989).

If an excess amount of organics is present in the influent, there will not be enough bacteria to completely consume these organics. Consequently, the excess organics will escape to the effluent water. The control of the organic loading is limited to reducing the influent flowrate. In the absence of temporary holding tanks, as in most treatment plants, the control of the incoming BOD₅ load is difficult.

2.1.1.10 Biological View

There are ample numbers of microorganism species present in the raw wastewater. Some of the major types are bacteria, fungi, algae, and protozoa. By constant maintenance of adequate organic substrates and

dissolved oxygen, the most suitable bacteria for the activated sludge process can be enriched.

The importance of keeping bacteria healthy is one of the most challenging tasks in the treatment plant operations. In order to achieve an efficient system, the bacteria needs to be kept under optimum conditions. Each bacterial species has a different environmental condition requirement. In wastewater the optimum conditions are satisfied by supplying suitable temperature, pH, nutrients, and oxygen (see Table 2-4).

The temperature has an effect on the speed of the biological reactions. Each species has its own optimum temperature. Because activated sludge water temperature varies within 10 to 22°C (Tchobanoglous and Schroeder, 1985), the most dominant bacteria fall in *psychrophilic*, and *mesophilic* regions.

The pH is another important factor in the biological processes. Many bacteria function properly in a very narrow pH range. If there is any acidic or basic chemical in the water, it will cause dramatic changes in the pH and prevent proper functioning of the bacteria.

The nutrients and oxygen are also vital requirements in the biological system. Not only will the bacteria not survive without nutrients or oxygen, but it needs precisely the proper amount of each to live.

Table 2-4: Optimum condition requirements for the microorganisms in the activated sludge systems for organic compound removal and for nitrification

Variables	Optimum values from literature		Reference
	Organic removal	Nitrification	
Dissolved Oxygen (mg/L)	0.7 - 1.5	≥ 1.0*	WPCF, 1987
pH	6.5 - 7.5	7.5 - 8.6	Metcalf and Eddy, 1991
Carbon Source	Biodegradable Organic Carbon	CO ₂	WPCF, 1990
Temperature (°C)	25 - 35*	15 - 35	WPCF, 1987
Inorganic Compounds	Na ⁺ , K ⁺ , Ca ⁺² , Mg ⁺² , PO ₄ ³⁻ , Cl ⁻ , SO ₄ ⁻ , Trace Metals		WPCF, 1977

* from Metcalf and Eddy, 1991

2.1.1.11 Nutrient Requirements

To keep microorganisms functioning properly, sufficient amounts of food for them must be available. The bacteria uses organics as both an energy and a carbon source. During respiration and synthesis, they use the carbon which is in the structure of the organic compound. In order for them to consume these organics, compounds have to be biodegradable.

In addition to organic compounds, inorganic chemicals which are listed in Table 2-4 are needed for the structure of the cell mass. If all the

parameters are supplied for the bacteria, then it grows and consumes organics efficiently in the wastewater. Consequently, the bacteria provides a good quality effluent discharge water.

2.1.1.12 Oxygen Requirements

The microorganisms in the activated sludge systems are strict aerobes. If the dissolved oxygen concentration drops below the bacteria's compensation limits, the bacteria can no longer survive. The lack of oxygen in the water causes undesirable anaerobic conditions. Once the oxygen is absent, different bacteria cultures dominate the population.

The oxygen levels in the aeration tanks are observed by DO probes. These probes are connected to the main control room gauges so the operator can monitor the O₂ concentration 24-hours a day. The rate of oxygen removal from mixed liquor by the microorganisms is called *oxygen uptake rate* (OUR). Based on the OUR test (see APHA *et al.*, 1989), the activated sludge plant can be operated efficiently (Stenstrom and Andrews, 1979; WPCF, 1989).

2.1.1.13 Common Microorganisms

Activated sludge is a mixed culture microorganism system. A large classification of microorganisms is present in the basins. Major categories

are as follows: bacteria, fungi, algae, protozoa, and viruses. Most common organisms are mesophilic and their sizes vary widely but generally range between 0.5 to 3 μm . Higher species are also frequently observed in activated sludge systems such as flagellates, ciliates (free swimming/crawling), and rotifers.

2.1.1.14 Filamentous Microorganisms

In fact not all microorganisms in activated sludge mixed-liquor are beneficial. There are viruses and bacteria in the water that can cause serious diseases. Some types of viruses are virulent and infectious. Chlorination of the effluent destroys these harmful organisms at approximately a 99% efficiency rate. In some cases, however, there is the possibility of a public health hazard caused by the escape of these virulent organisms.

There are other organisms that also cause problems in the process. *Filamentous* organisms cause swelling of floc by increasing the water content of the bacterial cell. In this way the floc particle will have similar density as the water itself and particles will not settle efficiently. This type of sludge which has a reduced settling capacity is called *bulking sludge*. This is one of the most common problems observed in the activated sludge process. If sludge bulking occurs and corrective actions are not taken, effluent quality may dramatically drop and excessive amounts of suspended matter may be discharged into the environment.

The issue of the bulking sludge problem is discussed in section 2.1.3.4 in detail.

2.1.1.15 Advanced Options: Mixing, Flexibility

Many problems in the activated sludge plants are caused by inadequate design or inefficient operation and maintenance. Mixing in the activated sludge basins is very crucial in obtaining a well-mixed tank. If there are any areas of a tank not well mixed, the oxygen concentration of this area will be lower than the rest of the tank. The anoxic or even anaerobic conditions may exist in the poorly mixed areas. Inadequate mixing may be caused by improper design or maintenance.

The activated sludge tanks which have diffusers need to be cleaned approximately once or twice a year. Failure to clean the tanks and diffusers will cause clogging of the diffusers and irregular mixing patterns in the basin.

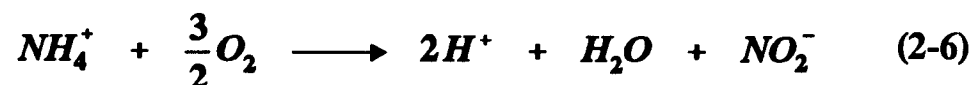
Flexibility is another important feature in the operations of the activated sludge process. The operator should be able to change different modifications of the system easily (*e.g.*, step feed to conventional, or high rate to contact-stabilization). Once the operator is able to change the system, he/she can try different procedures and select the most suitable one for that plant use.

Most of the problems are caused by the lack of flexibility. For instance, the operator cannot supply enough oxygen to the aeration tanks because there is a limitation on the maximum number of blowers which can be put on-line for the same aeration basin. Such simple situations may cause great damage to the process performance.

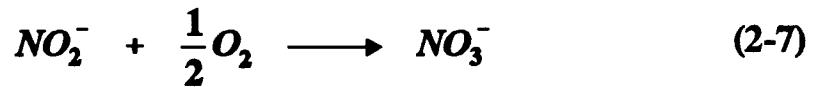
2.1.2 Nitrification Process Description

Nitrification is a process in which autotroph microorganisms convert ammonia (NH_4^+) to nitrate (NO_3^-) through a series of biochemical reactions. Nitrifying bacteria first isolated in the mid-nineteenth century by S. Winogradsky (1890) under strictly inorganic media. The findings of his work proved that the nitrification process is a biological conversion process of ammonia to nitrate by organisms which are chemoautotrophic, meaning that organisms can grow in a strictly inorganic media, utilizing CO_2 and inorganic materials to obtain carbon and metabolic energy. The nitrifying bacteria is also found to be obligate aerobes.

The conversion of ammonia to nitrate is achieved in two steps. In the first step, ammonia is oxidized to nitrite (NO_2^-) by a gram-negative, rod shaped bacteria, namely *Nitrosomonas* by the following equation:



The second step in the nitrification process is when the nitrite is converted to nitrate by *Nitrobacter* which has similar properties of *Nitrosomonas*. The conversion reaction is shown in the following equation:



Both above equations are energy yielding reactions where the produced energy is used during metabolic activities such as cellular growth and maintenance. Oxygen is a requirement for the above equations since it is the only electron acceptor in both reactions. Thus, presence of molecular oxygen is a mandatory condition in order to achieve the nitrification process.

The growth of nitrifying bacteria obeys the substrate controlled growth kinetics and it is highly influenced by pH, temperature, BOD₅/TKN ratio, ammonia, nitrite, and dissolved oxygen concentrations (EPA, 1975).

2.1.2.1 Nitrification Microbiology

In the mid-nineteenth century, the evidence of ammonia conversion to nitrate by a biological nitrification process drew the attention of leading microbiologists S. Winogradsky (1856-1953) and M. W. Beijerinck (1851-1931). Both investigators worked on cycles of carbon, nitrogen, and sulfur. Much of the work of Winogradsky focused on the nitrogen cycle.

In 1890, he discovered a new mode of life in the microbiology world, the chemoautotrophic bacteria. These type of bacteria can grow in strictly inorganic media in the presence of carbon dioxide. The energy for cellular growth and maintenance is obtained from the oxidization of the reduced inorganic compounds. The carbon in the cellular composition is obtained from the utilization of carbon dioxide.

Nitrification is a two-step process with the ammonia first being converted to nitrite and then to nitrate. Schematic representation of the nitrification process is shown in Figure 2-2.

The nitrification process can be described as a conversion of ammonia to nitrite by *Nitrosomonas* and to nitrate by *Nitrobacter*. Nitrate is further transformed to nitrogen gas (N_2) by the addition of organic carbon during the denitrification process (see Figure 2-2). The denitrifying bacteria is completely different from the nitrifying bacteria and it is out of the scope of this dissertation. The denitrification process is extensively studied elsewhere (Van Haandel *et al.*, 1981; Beccari *et al.*, 1983; Abufayed and Schroeder, 1986; Shieh and Mulcahy, 1986; Clayton *et al.*, 1991).

Part of ammonia consumed by the bacteria is used for cell synthesis and maintenance. During the cell synthesis process, ammonia is utilized by the assimilation process for the cell reproduction. Nitrogen is extracted from ammonia and used within the cell composition. When the cells die, the lysis process releases the nitrogen, which was bounded in the cell composition, into the system to be used as a nitrogen source by the living

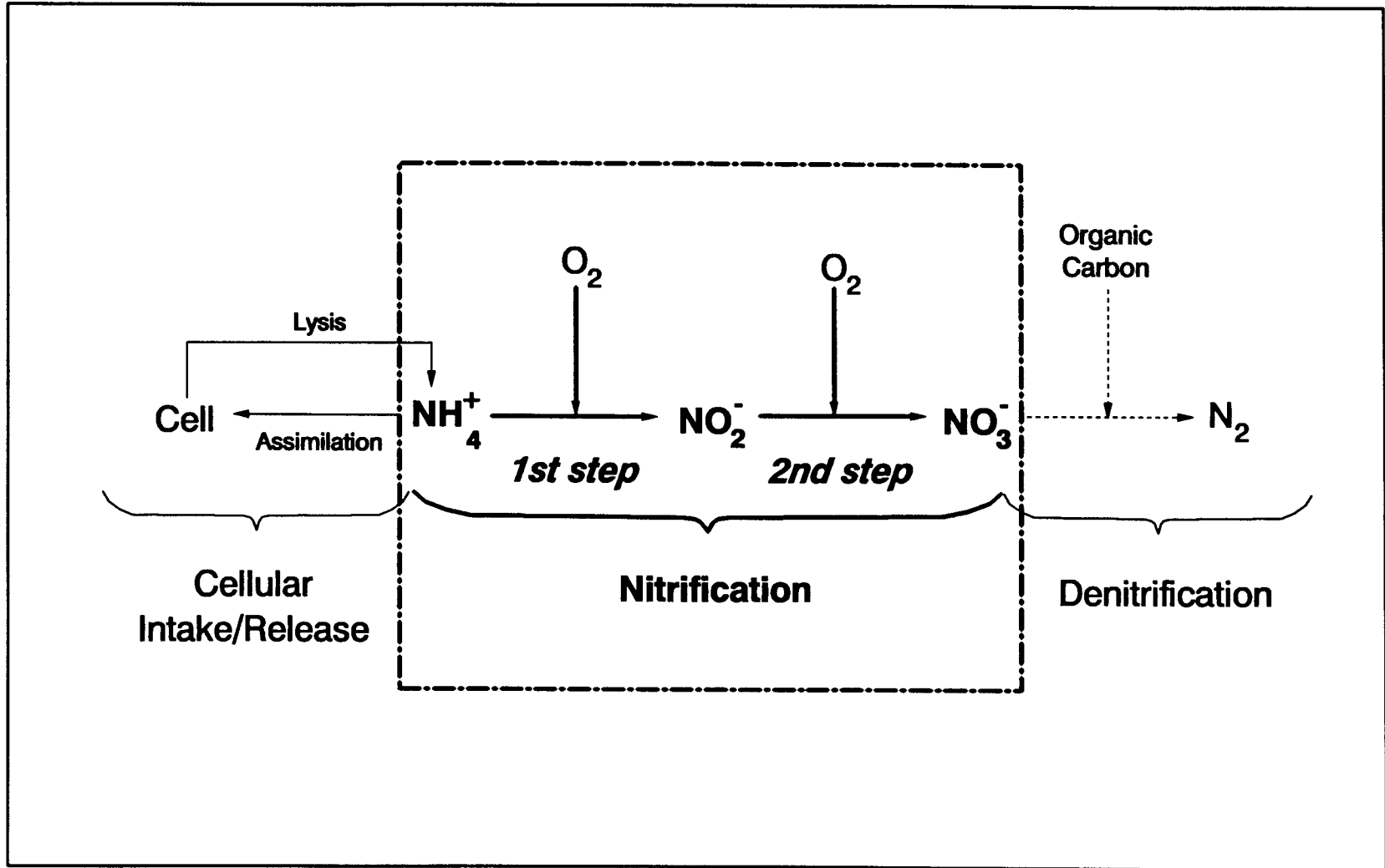
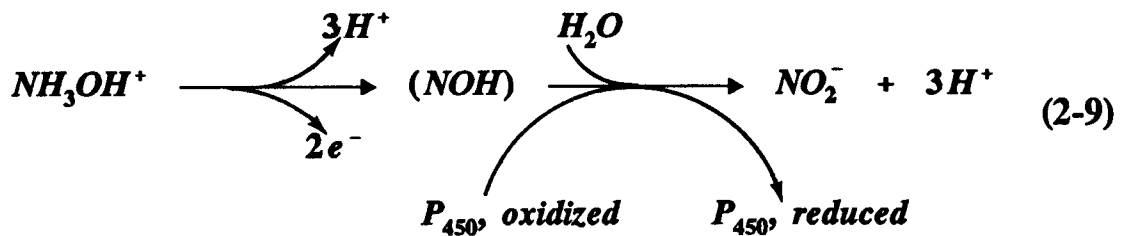
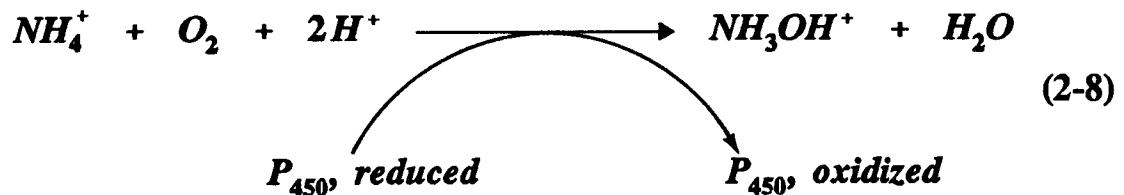


Figure 2-2: Nitrogen conversion by microorganisms

bacteria.

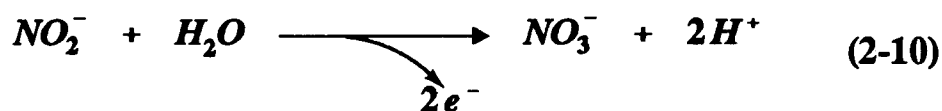
In the first step of the nitrification process, evidence of the existence of intermediary reactions was first suggested by Hoffman and Lees (1953) and later was supported by Engel and Alexander (1958). Works of Verstrate and Alexander (1972) and Hollocher *et al.*, (1981) suggested the use of molecular oxygen as an electron acceptor during the first stage of ammonia conversion. The suggested pathway of ammonia conversion to nitrite by a series of biochemical reactions is shown below (Stainer *et al.*, 1986):



In Eq. 2-8, ammonia is converted to hydroxylamine by oxidizing the cytochrome P₄₅₀ during the process. Hydroxylamine is the first intermediary step in the ammonia conversion process. In Eq. 2-9, hydroxylamine is further oxidized to nitrite possibly through the nitroxyl (NOH) step which is the second intermediary step. Two electrons are

released during the oxidation process and they are used during the ATP-generation cycle. The oxidation of nitroxyl to nitrite is mediated by the reduction of the P₄₅₀ and by an electron acceptor, possibly cytochrome *c*.

The second step in the nitrification process which consists of oxidation of nitrite to nitrate is a step similar to the ammonia oxidation reactions. Works of Aleem *et al.*, (1965) have demonstrated the source of oxygen in nitrate to be water and not the molecular oxygen. The suggested pathway for oxidation of nitrite to nitrate is illustrated by the following reaction (Stainer *et al.*, 1986):



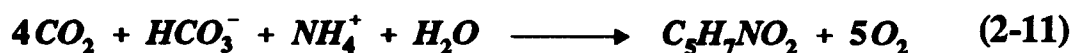
Because the oxidation of nitrite to nitrate is a less complex reaction than the ammonia conversion reactions, it is more easily understood. In Eq. 2-10, two electrons are released during the oxidation reaction where possibly cytochrome *a* is used as an electron acceptor for the nitrite oxidation reaction.

Microorganisms in the genera of nitrifying bacteria are a small group and they are categorized by several distinctive properties. Obligate autotrophy is a rule among the nitrifying bacteria with the exception of some strains of *Nitrobacter*. The form of the cells, however, may differ between rod, sphere, long rod, and spiral shaped forms and may have polar, subpolar, or peritrichous flagella. Most strains are observed to be motile and

reproduction is by binary fission for all the nitrifying bacteria but *Nitrobacter*. The *Nitrobacter* genus reproduces by the budding process.

In addition to *Nitrosomonas* and *Nitrobacter*, the genera of the nitrifying bacteria group includes ammonia conversion to nitrite microorganisms: *Nitrospira*, *Nitrosococcus*, *Nitrosolobus*, and nitrite to nitrate organisms: *Nitrospina*, and *Nitrococcus* (Stainer *et al.*, 1986). Table 2-5 shows the individual characteristics of these nitrifying bacteria.

All of the nitrifying bacteria are chemoautotroph which are able to extract carbon from carbon dioxide under aerobic conditions and the presence of a buffering agent and a sufficient amount of ammonia concentration. The cell synthesis can be illustrated by the following reaction (Metcalf and Eddy, 1991):



The empirical formula $\text{C}_5\text{H}_7\text{NO}_2$ is widely adopted in the literature to describe the cellular composition of the nitrifying bacteria (Metcalf and Eddy, 1991).

In addition to macronutrients such as carbon dioxide, oxygen, ammonia for *Nitrosomonas*, and nitrite for *Nitrobacter*, the presence of required micronutrients is also mandatory during the cell synthesis. Only trace amounts of these micronutrients are required. Excessive amounts of most of these micronutrients have inhibitory effects on the growth of the

Table 2-5: Some genera of nitrifying bacteria (Stainer/Ingraham/Wheelis/Painter, THE MICROBIAL WORLD, 5e, ©1986, p.385. Reprinted by permission of Prentice Hall, Englewood Cliffs, New Jersey)

Reaction	Cell Form	Flagella	Intracytoplasmic membrane	DNA Base Composition G+C %	Obligate Autotrophy	Genus
$\text{NH}_4^+ \rightarrow \text{NO}_2^-$	Rod	Subpolar	Lamellar	47 - 50	+	<i>Nitrosomonas</i>
	Tight spiral	Peritrichous	None	54	+	<i>Nitrospira</i>
	Sphere	Peritrichous	Lamellar	50 - 51	+	<i>Nitrosococcus</i>
	Irregular, lobed	Peritrichous	Vesicular	54 - 55	+	<i>Nitrosolobus</i>
$\text{NO}_2^- \rightarrow \text{NO}_3^-$	Rod, often pear-shaped	Polar	Lamellar	60 - 62	+ or -	<i>Nitrobacter</i>
	Long, slender rod	---	None	58	+	<i>Nitrospina</i>
	Sphere	Polar	Tubular	61	+	<i>Nitrococcus</i>

nitrifying bacteria. Table 2-6 shows some of the required micronutrients for *Nitrosomonas* and *Nitrobacter* and the required concentrations for each bacteria.

Phosphate, magnesium, iron, and copper are shown to be mandatory micronutrient requirements for the growth of both *Nitrosomonas* and *Nitrobacter* (Painter, 1977). The positive stimulatory effects of sodium salts on *Nitrosomonas* are also indicated by the works of Vargues and Brisou (1963) and Loveless and Painter (1968). Moreover, the work of Lees (1954) as cited in Stenstrom and Ng (1984) showed that calcium has a positive effect on the growth of *Nitrobacter*.

The limiting factors of nitrification have been widely studied in the literature. Major limiting factors for the nitrification process have been identified as pH, temperature, and dissolved oxygen concentration (Wild *et al.*, 1971; Charley *et al.*, 1980; Stenstrom and Poduska, 1980; Painter and Loveless, 1983; Randall and Buth, 1984a; Antoniou *et al.*, 1990; Hanaki *et al.*, 1990; Payraudeau *et al.*, 1990). It has been shown that the optimum pH range is 7.5 - 8.6 (WPCF, 1983); the optimum temperature range is 15 to 35°C (Wild *et al.*, 1971); and the limiting dissolved oxygen concentration is 1 mg/L (WPCF, 1983). The limiting effects of these parameters will be discussed in section 2.1.2.3 under nitrification in the activated sludge process.

The inhibition of nitrification by chemical exposure is also comprehensively researched (Meiklejohn, 1954; Tomlinson *et al.*, 1966;

Table 2-6: Some of micronutrients required for nitrifying bacteria

Micronutrient	Concentration (mg/L)	Required for		Reference
		<i>Nitrosomonas</i>	<i>Nitrobacter</i>	
Phosphate	310.0	Growth	Growth	Van Droogenbroeck and Laudelout, 1967
Magnesium	12.5 - 50.0	---	Growth	Loveless and Painter, 1968
Iron	0.5 - 0.6	Growth	---	Skinner and Walker, 1961
Copper	0.1 (in activated sludge)	Activity	---	Tomlinson <i>et al.</i> , 1966
	0 - 0.06	Activity	---	Loveless and Painter, 1968
Sodium	1.5 - 7.0	Growth	---	Loveless and Painter, 1968
Calcium	---	---	Growth	Lees, 1954

Loveless and Painter, 1968; Anthonisen *et al.*, 1976; Sujarittanonta and Sherrard, 1981). Among the many other paths, the inhibitors are shown to be effective via destruction of cell metabolism such as the inhibition of cell synthesis, and via inhibition of oxidation reactions. The inhibitors have more significant effects on the oxidation of ammonia to hydroxylamine by *Nitrosomonas* than on the reaction by *Nitrobacter*.

Chemicals that are found in refinery wastewater can be toxic to the nitrifying bacteria in the concentrations shown in Table 2-7. In this dissertation, the term toxic chemical refers to the chemicals which have toxic or inhibitory effect on the nitrifying bacteria. Table 2-7 combines the parameters shown in Table 2-2 and the toxicity experiments conducted in the literature using these chemicals. Table 2-7 lists the chemicals in the descending order in which they are found in the refinery treatment plant influent.

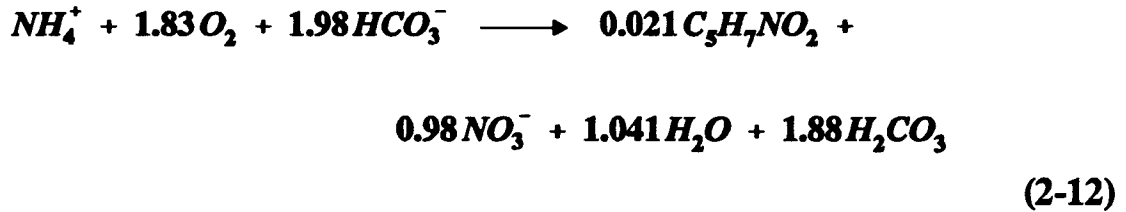
2.1.2.2 Nitrification Stoichiometry and Kinetics

Nitrification is a process in which the inorganic nitrogen is oxidized to obtain energy for cell synthesis of chemoautotroph nitrifying bacteria. Equations 2-6 and 2-7 show the reactions for ammonia conversion to nitrate. These equations produce energy which is used for cell synthesis as shown in Eq. 2-11. EPA (1975) has suggested combining ammonia oxidation to nitrate and the cell synthesis by the following reaction:

Table 2-7: Some of the chemicals which are found in the refinery treatment plant influent, their inhibitory concentrations, and inhibitory effects on the nitrifying bacteria

Compound	Concentration		Inhibitory Effect (%)	Reference
	Refinery Influent* (mg/L)	Inhibitory Level (mg/L)		
Ammonia	75.0	0.1 - 1.0 (<i>Nitrobacter</i>)	substantial	Anthonisen <i>et al.</i> , 1976
		> 10 (<i>Nitrosomonas</i>)	substantial	
Phenol	1.5	15.0 - 20.0	50	Neufeld <i>et al.</i> , 1986
		14.4 (2.55)	50 (no effect)	Pantea-Kiser <i>et al.</i> , 1990
		5.6	75	Stafford, 1974
Zinc	0.36	0.08 - 0.5	some	Loveless and Painter, 1968
		0.01 M	substantial	Meiklejohn, 1954
Cyanide	0.15	0.65	75	Tomlinson <i>et al.</i> , 1966
		> 0.5	substantial	Richardson, 1985
		0.068 (0.013)	50 (no effect)	Pantea-Kiser <i>et al.</i> , 1990
Arsenic	0.12	0.013 M (Na Arsenite)	75	Tomlinson <i>et al.</i> , 1966
Copper	0.065	0.05 - 0.56	substantial	Loveless and Painter, 1968
		4	75	Tomlinson <i>et al.</i> , 1966
		0.01 M	substantial	Meiklejohn, 1954
Nickel	0.02	0.0002 M	100	Meiklejohn, 1954
		0.0004 - 0.00004 M	75	Tomlinson <i>et al.</i> , 1966
		5 - 15	100	Randall and Buth, 1984b
Nitrous Acid	---	0.22 - 2.8	substantial	Anthonisen <i>et al.</i> , 1976

* Vernick *et al.*, 1984



Equation 2-12 estimates the oxygen requirements, the alkalinity consumption, and the nitrifier biomass production rate during the nitrification process. Equation 2-12 suggests that for each mole of NH_4^+ , 1.83 moles of oxygen and 1.98 moles of alkalinity is required. In other terms, 4.3 mg of O_2 and 7.14 mg of alkalinity (as CaCO_3) is required to convert a 1 mg of $\text{NH}_4^+\text{-N}$. Furthermore, 0.021 moles of biomass is produced per one mole of NH_4^+ converted (or 0.17 mg of cells per mg of $\text{NH}_4^+\text{-N}$ used).

Equation 2-12 provides valuable information for designing wastewater treatment plants. For example, the oxygen requirement is used to design the air supply systems, and the alkalinity consumption is used to control the pH value. The optimum pH is difficult to obtain because it falls within a narrow range for the nitrification process. And, finally, the biomass production rate is used in the process control of sludge age and sedimentation tank operations.

Kinetics of the nitrification process can be expressed by the substrate limiting formula which is proposed by Monod (1949) as shown below:

$$\mu = \mu_m \frac{S}{K_s + S} \quad (2-13)$$

where μ = Specific growth rate, 1/d
 μ_m = Maximum specific growth rate, 1/d
 S = Limiting substrate concentration, mg/L
 K_s = Half-saturation constant, mg/L

Assuming oxygen is not a limiting factor, $\text{NH}_4^+\text{-N}$ becomes the limiting substrate in the nitrification process because the second step (nitrite to nitrate) is not going to take place unless ammonia is converted to nitrite first. In other words, if the growth of *Nitrosomonas* is inhibited because the nitrification process is a sequential reaction, the growth of *Nitrobacter* will also be inhibited. In contrast, inhibition of the *Nitrobacter* growth will have little effect on the ammonia conversion process. Thus, the nitrification kinetics can be expressed with the following equation:

$$\mu_A = \mu_{m,A} \frac{S_{NH}}{K_{NH} + S_{NH}} \quad (2-14)$$

where μ_A = Specific growth rate of autotrophs (*Nitrosomonas*), 1/d
 $\mu_{m,A}$ = Maximum specific growth rate of autotrophs, 1/d
 S_{NH} = Ammonia concentration, mg $\text{NH}_4^+\text{-N/L}$
 K_{NH} = Half-saturation constant for NH_4^+ , mg/L

Equation 2-14 expresses the relationship between the specific growth rate and the $\text{NH}_4^+\text{-N}$ concentration by both first and zero order kinetics depending on the $\text{NH}_4^+\text{-N}$ concentration. When the $\text{NH}_4^+\text{-N}$ concentration

is much smaller than K_{NH} constant which can happen at low NH_4^+ -N concentrations, Equation 2-14 displays first order kinetics. However, when the NH_4^+ -N concentration becomes much larger than K_{NH} constant at high NH_4^+ -N concentrations, the equation exhibits zero order reaction kinetics. The transition between first and zero order reaction kinetics is illustrated in Figure 2-3.

The growth rate of autotrophs can be quantified by the following expression:

$$r_A = \mu_A X_A \quad (2-15)$$

where r_A = Rate of autotrophic bacteria growth, mg *Nitrosomonas*/L·d
 X_A = Concentration of autotrophic bacteria, mg *Nitrosomonas*/L

The ammonia conversion rate is thus expressed in terms of the growth rate of *Nitrosomonas* by the following equation:

$$\frac{dS_{NH}}{dt} = r_N = - \frac{r_A}{Y_A} \quad (2-16)$$

where r_N = NH_4^+ -N utilization rate, mg NH_4^+ -N/L·d
 Y_A = Maximum yield coefficient, mg autotrophic biomass produced/mg NH_4^+ -N consumed

By combining Eq. 2-15 and 2-16, the expression for ammonia utilization rate becomes:

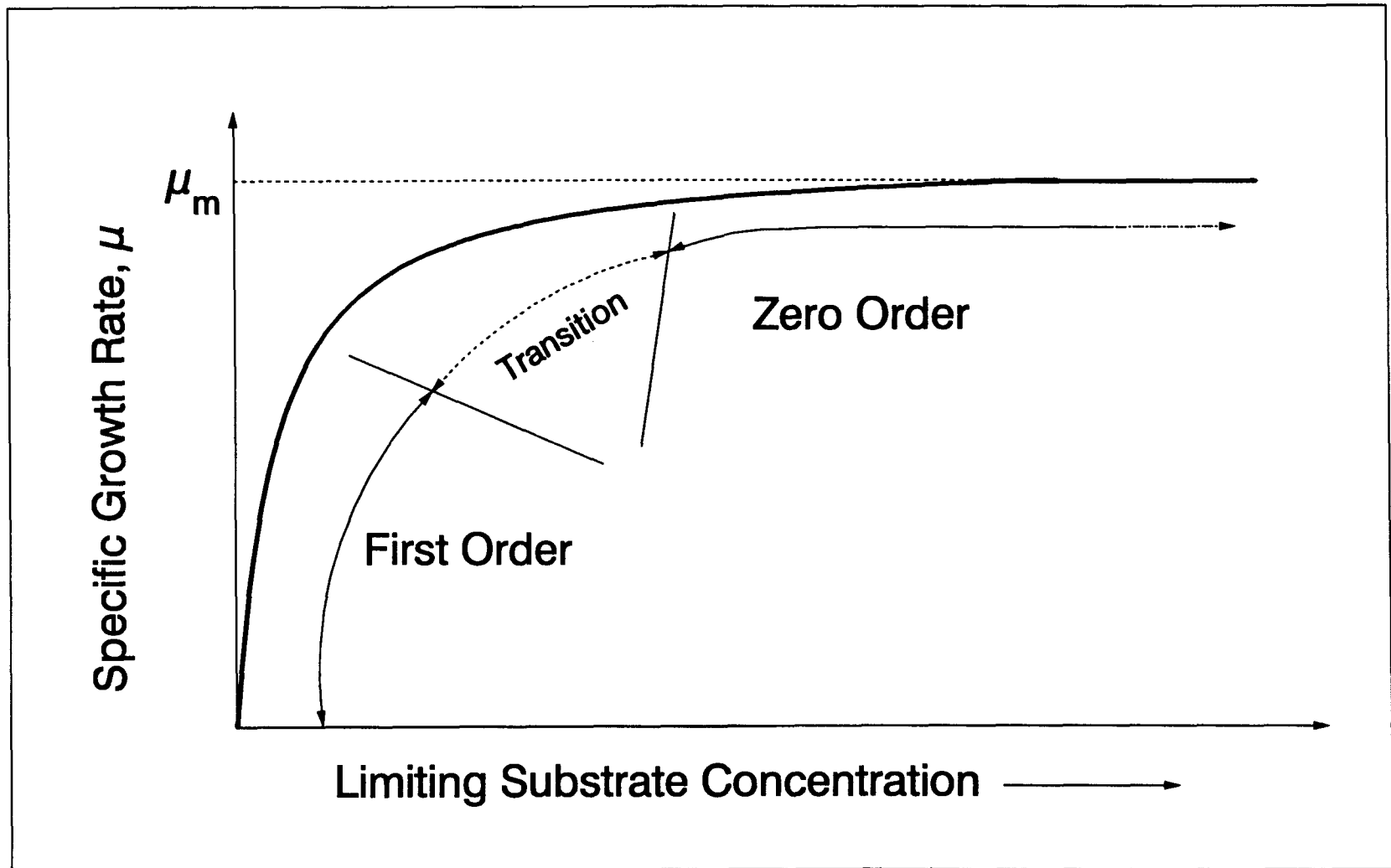


Figure 2-3: First and zero order reactions in substrate-limited growth kinetics

$$r_N = - \mu_A \frac{X_A}{Y_A} \quad (2-17)$$

Incorporating Eq. 2-14 into Eq. 2-17 yields the following expression for the amount of ammonia removed per liter per day during the nitrification process:

$$r_N = - \mu_{m,A} \frac{S_{NH}}{Y_A (K_{NH} + S_{NH})} X_A \quad (2-18)$$

Studies have indicated that the ammonia concentration is not the only limiting substrate in nitrification kinetics. Dissolved oxygen plays a significant role in the growth rate of autotrophic bacteria (Charley *et al.*, 1980; Stenstrom and Poduska, 1980; Hanaki *et al.*, 1990). The limiting factor of dissolved oxygen can be expressed as shown below:

$$\mu = \mu_m \frac{S_O}{K_O + S_O} \quad (2-19)$$

where S_O = Concentration of dissolved oxygen, mg/L

K_O = Half-saturation concentration for dissolved oxygen, mg/L

Adding the oxygen limiting substrate term in Eq. 2-19 to Eq. 2-14 demonstrates the existence of two limiting substrates, ammonia and dissolved oxygen concentration as shown below:

$$\mu_A = \mu_{m,A} \frac{S_{NH}}{K_{NH} + S_{NH}} \cdot \frac{S_O}{K_{O,A} + S_O} \quad (2-20)$$

where $K_{O,A}$ = Dissolved oxygen saturation constant for autotrophic bacteria, mg/L

Similarly, if the same development methodology for Eq. 2-15 through 2-18 is applied to Eq. 2-20, the following expression is obtained for the ammonia reduction rate:

$$r_N = - \mu_{m,A} \frac{S_{NH}}{Y_A (K_{NH} + S_{NH})} \cdot \frac{S_O}{K_{O,A} + S_O} X_A \quad (2-21)$$

The coefficients in Eq. 2-21, which are $\mu_{m,A}$, K_{NH} , $K_{O,A}$, and Y_A , have to be determined for each individual plant so that the Eq. 2-21 describes that plant accurately. Although there is a comprehensive work in the literature relating to the estimation of these parameters, there is no accurate method known to set values for these coefficients.

Previous work in the literature revealed a wide range of values for these kinetic constants depending on the experimental conditions such as, pH, DO, temperature, and the origin of the bacterial culture (*e.g.*, pure culture or activated sludge culture). Furthermore, these coefficients are different for *Nitrosomonas* and *Nitrobacter*, thus, individual experiments are needed to determine the coefficients for each nitrifying bacteria. Table 2-8 and Table 2-9 show the typical values of the kinetic coefficients reported in the literature for *Nitrosomonas* and for *Nitrobacter*, respectively.

Both Table 2-8 and 2-9 indicate that there is wide range of values for each coefficient and there is no set method to determine the values for these coefficients. The range of values for each coefficient is compiled in Table 2-10(a) and (b) from Table 2-8 and 2-9.

Table 2-8: Kinetic coefficients reported in the literature for *Nitrosomonas*

Culture	Temperature	$\mu_{m,A}$	K_{NH}	Y_A	pH	Reference	
Pure	< 20° C	0.52			7.5	Painter and Loveless, 1983	
		0.25 - 0.50				Buswell <i>et al.</i> , 1954	
	≥ 20° C	2.2	10	0.06	7.0 - 7.4	Skinner and Walker, 1961	
		0.66 [†]	1.0 [†]	0.38 [*]	7.3 - 7.8	Hanaki <i>et al.</i> , 1990	
		0.72 [‡]	1.1 [‡]	0.68 [*]	7.3 - 7.8		
		0.25	0.8	0.12	7.0	Rozich and Castens, 1986	
		1.38	1.0	0.03 - 0.1	8.0	Loveless and Painter, 1968	
Activated sludge	< 20° C	0.41 - 0.59			7.5 - 8.5	Painter and Loveless, 1983	
		0.120 - 0.317 [*]			6.8 - 8.2	Antoniou <i>et al.</i> , 1990	
			0.61 - 1.58		7.5	Charley <i>et al.</i> , 1980	
	≥ 20° C	1.08	0.063		7.2	Poduska and Andrews, 1975	
		0.33	1.0	0.05	7.5 - 8.0	Downing <i>et al.</i> , 1964	
		0.32			7.3	Hall and Murphy, 1980	
		0.45		0.15	7.5 - 8.0	Beccari <i>et al.</i> , 1979	
		0.1 - 0.36			8.5	Painter and Loveless, 1983	
		0.12 - 0.78			7.0 - 8.0		
		0.14 - 0.48			6.5		
			1.54 - 2.50		7.5	Charley <i>et al.</i> , 1980	
					0.22	Argaman and Brenner, 1986	
		0.564 - 0.971 [*]			8.0 - 8.5	Antoniou <i>et al.</i> , 1990	
		0.298 - 0.638 [*]			6.4 - 6.9		
					0.29 [□]	8.0	Gee <i>et al.</i> , 1990a
					0.36 [*]	8.0	
					0.30 [□]	8.0	Gee <i>et al.</i> , 1990b
0.72 - 0.78 [‡]	1.1 - 9.4 [‡]			7.3 - 7.8	Hanaki <i>et al.</i> , 1990		
0.69 [†]	1.0 [†]			7.3 - 7.8			

[†] Sufficient DO; [‡] DO = 0.5 mg/L; ^{*} Growth yield index (1/d); ^{*} Values reported for ($\mu_{m,A}$ -b_A); [□] mg VSS/mg NH₄⁺-N; ^{*} mg VSS/mg N

Table 2-9: Kinetic coefficients reported in the literature for *Nitrobacter*

Culture	Temperature	$\mu_{m,A}$	K_{NH}	Y_A	pH	Reference	
Pure	< 20° C		1.4 - 2.1			Laudelot and Van Trichelon, 1960	
	≥ 20° C		5	0.04 - 0.07		Gould and Lees, 1960	
		1.39		0.02	7.7	Boon and Laudelot, 1962	
Activated Sludge	≥ 20° C	1.44	≤ 0.16		7.2	Poduska and Andrews, 1975	
		0.27		0.07	7.5 - 8.0	Beccari <i>et al.</i> , 1979	
		0.26			8.5	Painter and Loveless, 1983	
		0.1 - 0.32			7.0 - 8.0		
		0.22 - 0.45			6.5		
					0.084 [†]	8.0	Gee <i>et al.</i> , 1990a
					0.0831 [†]	8.0	Gee <i>et al.</i> , 1990b

[†] mg VSS/mg NO₃-N

Table 2-10(a): Typical range of kinetic coefficient values for *Nitrosomonas* and *Nitrobacter* under various temperatures in the pure cultures (pH = 7.0-8.0)

Temperature	$\mu_{m,A}$		K_{NH}		Y_A	
	<i>Nitrosomonas</i>	<i>Nitrobacter</i>	<i>Nitrosomonas</i>	<i>Nitrobacter</i>	<i>Nitrosomonas</i>	<i>Nitrobacter</i>
< 20° C	0.25 - 0.52	---	---	1.4 - 2.1	---	---
≥ 20° C	0.25 - 1.38	1.39	0.8 - 10	5	0.03 - 0.68	0.02 - 0.07

Table 2-10(b): Typical range of kinetic coefficient values for *Nitrosomonas* and *Nitrobacter* under various temperatures in the activated sludge cultures (pH = 6.4-8.5)

Temperature	$\mu_{m,A}$		K_{NH}		Y_A	
	<i>Nitrosomonas</i>	<i>Nitrobacter</i>	<i>Nitrosomonas</i>	<i>Nitrobacter</i>	<i>Nitrosomonas</i>	<i>Nitrobacter</i>
< 20° C	0.1201 - 0.59	---	0.61 - 1.58	---	---	---
≥ 20° C	0.1 - 1.08	0.1 - 1.44	0.063 - 9.4	≤ 0.16	0.05 - 0.36	0.07 - 0.084

The reasons for reporting a wide range of values for the kinetic coefficients might be due to the absence of standardized measuring techniques, the absence of accurate formulation of the bacterial growth kinetics, and the presence of an excessive number of unknown conditions affecting the nitrification process. Until standardized, more accurate, and better representative experiments are conducted, the estimation of the nitrification kinetic coefficients will be limited.

2.1.2.3 Nitrification in the Activated Sludge Process

Many wastewater treatment plants are required to nitrify in order to meet their NPDES permits. In most municipal treatment plants, in addition to the ammonia standard, the permit also restricts the maximum nitrate discharge. Consequently, many treatment plants are equipped to perform both the nitrification and the denitrification processes in order to reduce the ammonia and the nitrate effluent concentrations in their discharge water.

Past experience shows that nitrification in the activated sludge process is extremely sensitive to environmental conditions and to process control parameters (Wild *et al.*, 1971; Painter and Loveless, 1983; Antoniou *et al.*, 1990). In addition to the pH, temperature, and dissolved oxygen level, the sludge age is a key process control parameter which significantly affects the nitrification process in the activated sludge units. Furthermore, organic loading, hydraulic retention time, absence of required nutrients,

competition between heterotrophs and nitrifying bacteria, and the presence of toxic chemicals have an additional effect on the nitrification process in the activated sludge units.

In the activated sludge process, endogenous decay of bacteria becomes a significant factor in the bacteria growth kinetics because the bacteria are kept in the aeration basins for an extended amount of time. The decay factor in the nitrification process is significant because the residential time is longer for the nitrifying bacteria due to its slow growth rate. When part of the biomass dies, the active portion of the biomass is reduced. To take this reduction into account, the specific growth rate equation (Eq. 2-14) is modified as shown below:

$$\mu'_A = \mu_{m,A} \frac{S_{NH}}{K_{NH} + S_{NH}} - b_A \quad (2-22)$$

where μ'_A = Net specific growth rate for autotrophs, 1/d

b_A = Endogenous decay coefficient, 1/d

The overall decay coefficient for both *Nitrosomonas* and *Nitrobacter* is in the range of 0.03 to 0.06 d⁻¹ with a typical value of 0.05 d⁻¹ (Metcalf and Eddy, 1991).

Competition between the heterotrophic bacteria and the nitrifiers plays an important role in the nitrification process. The nitrifying bacteria has a much slower growth rate than the heterotrophic bacteria. Consequently, much of the bacterial population consists of the heterotrophic bacteria.

Only under optimum conditions do the nitrifying bacteria become a significant part of the activated sludge culture. For industrial applications such as refineries, these optimum conditions might be an MCRT exceeding 20 days, the pH equal 7.6, the DO equal or exceed 3.0 mg/L, and the temperature range from 25-35°C. However, even under the most favorable conditions, the fraction of nitrifiers rarely exceeds 30% of the population (EPA, 1975).

The fraction of nitrifiers present can be estimated by using the BOD₅/TKN (Total Kjeldahl Nitrogen) ratio method (EPA, 1975). TKN represents the total nitrogen concentration including the organic and ammonia nitrogen (see APHA *et al.*, 1989). In refinery wastewater, however, because the organic nitrogen concentration is negligible, the TKN is assumed to be equal to the ammonia concentration.

The BOD₅/TKN ratio method to estimate the fraction of nitrifiers in the activated sludge is based on the yield coefficients of both the heterotrophic and the autotrophic bacteria. The following equation is used to estimate the fraction of nitrifiers (EPA, 1975):

$$f = \frac{V X_A}{V X_A + V X_H} \quad (2-23)$$

where:

$$V X_A = Y_A (S_{NH,o} - S_{NH,e}) \quad (2-24)$$

$$V X_H = Y_H (S_{BOD,o} - S_{BOD,e}) \quad (2-25)$$

Assume $S_{NH,e} \approx 0$ and $S_{BOD,e} \approx 0$. Apply Eq. 2-25 and 2-24 into Eq. 2-23 and rearrange:

$$f = \frac{1}{\left(\frac{S_{BOD,o}}{S_{NH,o}}\right) \frac{Y_H}{Y_A} + 1} \quad (2-26)$$

where

- f = Fraction of nitrifiers
- X_H = Heterotroph concentration, mg/L
- $S_{NH,o}$ = NH_4^+ -N concentration in the influent, mg/L
- $S_{NH,e}$ = NH_4^+ -N concentration in the effluent, mg/L
- Y_H = Maximum yield coefficient for heterotrophs
- $S_{BOD,o}$ = BOD_5 concentration in the influent, mg/L
- $S_{BOD,e}$ = BOD_5 concentration in the effluent, mg/L
- $S_{BOD,o}/S_{NH,o}$ = BOD_5 /TKN ratio

With Eq. 2-26 the fraction of nitrifiers and the BOD_5 /TKN ratio can be tabulated. Table 2-11 shows the estimation of nitrifier fraction based on the BOD_5 /TKN ratio using Eq. 2-26 when $Y_A = 0.15$ and $Y_H = 0.55$. From Table 2-11, it can be concluded that the nitrifier fraction increases with the increasing nitrogen content (or decreasing BOD_5 /TKN ratio).

In addition to the above mentioned factors, there are more variables affecting the nitrification process performance in the activated sludge units. These variables include the effects of MCRT, pH, temperature, dissolved oxygen concentration, organic loading, hydraulic retention time, absence of required nutrients, and the presence of toxic chemicals. These factors are discussed in order to understand the impact of each parameter on the nitrification process.

Table 2-11: The estimation of nitrifier fraction for given BOD₅/TKN ratios (EPA, 1975)

BOD ₅ /TKN ratio	Nitrifier fraction*
0.5	0.35
1	0.21
2	0.12
3	0.083
4	0.064
5	0.052
6	0.043
7	0.038
8	0.033
9	0.029

* Calculated using Eq. 2-26 ($Y_A = 0.15$; $Y_H = 0.55$)

Effects of MCRT. Mean cell retention time or sludge age is the key controlling parameter in achieving the nitrification process. Because the nitrifying bacteria has a slower growth rate than the carbon-consuming bacteria in the activated sludge, MCRT needs to be long enough to allow the nitrification to occur. Although it has been established that under laboratory conditions, the nitrification normally occurs when MCRT is between 2 to 3 days (Poduska, 1973), the real-time operations of activated sludge processes have proven that the minimum MCRT should be approximately 20 days in order for nitrification to take place (WPCF, 1987). Assuming the zero-order kinetics and steady state conditions, the

following expression is used to estimated MCRT using the specific growth rate:

$$MCRT = \frac{1}{\mu'_A} = \frac{1}{\mu_A - b_A} \quad (2-27)$$

The required MCRT can be obtained by adjusting the sludge wasting rate according to the methods discussed in section 2.1.4.3. Maintaining the required MCRT is a critical part of achieving a complete nitrification process.

Effects of pH. pH has a significant effect on the nitrification process. A required pH range for the nitrification process is a narrow range between 7.5 and 8.6 (WPCF, 1983). Several studies conducted on different pH ranges have demonstrated a similar optimum range for pH. Loveless and Painter (1968) have reported the optimum nitrification rate to be in the pH range of 7.0 to 8.2. A range of 7.8 to 9.8 with an optimum of 8.4 was reported by Wild *et al.* (1971). Additional confirming findings include a range of 7.5 to 8.5 with an optimum of 8.0 (Downing *et al.*, 1964; Painter and Loveless, 1983), and an optimum pH of 7.8 (Antoniou *et al.*, 1990).

It has been shown that the lowest threshold for nitrification to occur was between pH 6.0 and 6.5 (Painter and Loveless, 1983). Below pH 6.0, no significant nitrifying bacteria activity took place. Between pH 7.8 and 8.9, the maximum nitrification rate was observed to be 90 to 100% of the maximum rate (Wild *et al.*, 1971). The nitrification rate was decreased

to approximately 50% of the maximum rate when pH was lower than 7.0 or greater than 9.8, while pH greater than 10 completely inhibited the nitrifier activities (Downing *et al.*, 1964). Generally an optimum pH of 7.6 is recommended for the operations of activated sludge processes to achieve a complete nitrification.

One method to model the effects of pH on the nitrification process quantitatively is to establish a correlation between the specific growth rate and the pH value. The following equation is proposed by the EPA (1975) to account for the effect of pH on the growth kinetics of the nitrifiers.

$$\mu'_A = \mu_{m,A} [1 - 0.833 (7.2 - pH)] \quad (2-28)$$

where μ'_A = pH corrected specific growth rate of autotrophs, 1/d
 pH = pH of the wastewater

More complex models have been developed to include the pH variable in the nitrification kinetics. Antoniou *et al.* (1990) correlated pH, temperature, and the specific growth rate under various conditions. They proposed to use the following equation to account for the effect of pH and temperature in the autotrophic growth kinetics as shown below:

$$\mu_{m,A} - b_A = \frac{m e^{-(a/T)}}{1 + \frac{b}{10^{-pH}} + \frac{10^{-pH}}{c}} \quad (2-29)$$

where m, a, b, c = Parameters

T = Temperature of the wastewater, °K

After conducting a number of experiments under different temperatures and different pH values, the proposed correlation model constants were determined by using experiment results and fitting a line through the data points by the least squares method. The parameter values were determined as shown below:

$$m = 4.70 \times 10^{14}$$

$$a = 9.98 \times 10^3$$

$$b = 2.05 \times 10^{-9}$$

$$c = 1.66 \times 10^{-7}$$

In Eq. 2-29, pH gives an optimum value depending on the temperature. It has been reported by the authors that the optimum pH for the nitrification process is approximately equal to 7.8 for the feasible range of temperature values.

Effects of temperature. The temperature effects on the microorganisms may be due to the limitation on the mass transfer capacity, the effects on the reaction kinetics, and the changes in the microbial population composition (Characklis and Gujer, 1979).

As it is true for the general microbiology, every microorganism genera has its own optimum temperature range in which the growth and the activity are at the maximum rate. For the nitrifying bacteria, it has been reported that the optimum temperature range was between 30 to 35°C with a significant decrease in the activity and the growth rate outside the range of 18 to 35°C (EPA, 1975). Charley *et al.* (1980) reported increasing a

specific growth rate with the increasing temperature between 10 to 35°C with the exception of maximum ammonia oxidation at 15°C.

An increasing ammonia oxidation rate with the increasing temperature was also reported by Payraudeau *et al.* (1990) with approximately 0.5 kg NH₄⁺-N/m³·d removal rate at 6°C and approximately 1.0 kg NH₄⁺-N/m³·d removal rate at 20°C. Wild *et al.* (1971) reported an optimum temperature of 30°C with the decreasing nitrification rate as the temperature decreases to 5°C.

Studies conducted on the temperature effects showed that lower temperatures have greater impact on *Nitrobacter* than on *Nitrosomonas*. Thus, until the temperature decrease completely suppresses the nitrification process, nitrite accumulation may occur in the activated sludge processes (Randall and Buth, 1984a). The threshold temperature for the nitrite accumulation was identified as approximately 10°C. They also reported an increasing ammonia and nitrite oxidation from temperature 10 to 30°C while the nitrifying bacteria were very sensitive to the temperature changes between 10 to 17°C.

The formulation of the temperature effects on the nitrification kinetics is investigated by several researchers (EPA, 1975; Gujer, 1977; Painter and Loveless, 1983; Hall and Murphy, 1985; Payraudeau *et al.*, 1990). The findings showed that the temperature effect on the nitrification process simulates the Arrhenius law. Table 2-12 shows how the temperature effect

is taken into account for the nitrification kinetics and the reaction rates as proposed by different researchers.

In addition to Table 2-12, Antoniou *et al.* (1990) developed a model which takes both the pH and the temperature effects into account for the nitrification process kinetics. Eq. 2-29 shows the proposed correlation between the autotrophic specific growth rate, pH, and temperature.

Effects of dissolved oxygen. Oxygen is a requirement for the nitrification process. Eq. 2-12 estimates that 4.3 mg O₂ is needed to oxidize 1 mg of NH₄⁺-N. If a sufficient amount of DO is not available to the nitrifying bacteria, a decrease in the nitrification rate is observed. When DO is below the required levels, in addition to the ammonia concentration, the DO concentration is identified as a substrate limiting factor in the determination of the specific growth rate for the nitrifiers (see Eq. 2-20).

Moreover, lower oxygen concentrations are found to be an inhibitor to the nitrification process. Downing *et al.* (1964) observed inhibitory effects on the nitrifying bacteria when the DO was approximately 0.2 mg/L. Hanaki *et al.* (1990) also reported lower ammonia oxidation rates in the combined systems (*i.e.*, systems contain other types of bacteria besides the nitrifiers) under low DO (0.5 mg/L) conditions. However, they also reported contrasting findings from the experiments with the pure cultures of nitrifying bacteria which indicate no effect on the nitrification rate due to the low DO concentrations.

Table 2-12: Effect of temperature on the nitrification kinetic parameters and the reaction rates

Temperature Effect	Reference
<p><u>for Nitrosomonas</u> $\mu_A^* = \mu_{m,A} e^{0.098(T'-15)}$ $K_{NH}^* = 10^{0.051 T' - 1.158}$</p> <p><u>for Nitrobacter</u> $\mu_A^* = \mu_{m,A} e^{0.69(T'-15)}$ $K_{NH}^* = 10^{0.063 T' - 1.149}$</p>	EPA, 1975
<p><u>for Nitrosomonas</u> $\mu_A^* = \mu_{m,A} e^{0.11(T'-10)}$</p>	Gujer, 1977
$\mu_A^* = \mu_{m,A} e^{0.0729(T'-15)}$	Painter and Loveless, 1983
<p><u>for Nitrosomonas</u> $r_{max} = 1.67 \times 10^{10} e^{(-14,900 / RT)}$</p> <p><u>for Nitrobacter</u> $r_{max} = 1.102 \times 10^{12} e^{(-16,980 / RT)}$</p>	Hall and Murphy, 1985
$R_T = R_{20} 1.04^{(T'-20)}$ for $T' > 10^\circ\text{C}$	Payraudeau <i>et al.</i> , 1990

where

- μ_A^* = Temperature adjusted specific growth rate, 1/d
- T' = Temperature in °C
- K_{NH}^* = Temperature adjusted half-saturation constant for NH_4^+ , mg/L
- r_{max} = Nitrogen utilization rate, mg N/mg TSS · h
- R = Gas constant, cal/mole · °K
- T = Temperature in °K
- R_T = Reaction rate at temperature $T^\circ\text{C}$
- R_{20} = Reaction rate at temperature 20°C

Stenstrom and Song (1991) developed a model to investigate the relationship between DO and the nitrification process. The model consists of terms including the mass-transport resistance effects and the competition between the heterotrophic and the nitrifying bacteria. They reported that for the activated sludge process, the range of DO from 0.5 to 2.0 mg/L may become a limiting factor in the nitrification process depending on the MCRT and the resistance in the mass-transport. For pure cultures, they found a much lower DO limiting value of less than 0.1 mg/L. They also indicated that DO as high as 4.0 mg/L may be a limiting factor during a high organic loading due to the competition between the heterotrophs and the nitrifiers and the resistance in mass-transport mechanism.

Stenstrom and Poduska (1980) adapted the Monod kinetic equation to the nitrification process in the activated sludge process by taking DO into account as shown below:

$$\mu'_A = \mu_{m,A} \left[\frac{S_{NH}}{K_{NH} + S_{NH}} \cdot \frac{S_O}{K_{O,A} + S_O} \right] - b_A \quad (2-30)$$

If the oxygen concentration is below $K_{O,A}$, DO becomes a significant limiting factor in Eq. 2-30 (WPCF, 1983). The range of values for $K_{O,A}$ and b_A reported in the literature for *Nitrosomonas* and for *Nitrobacter* are listed in Table 2-13.

The effects of high DO levels are also investigated. Charley *et al.* (1980) reported that after feeding pure oxygen, the initial inhibition of nitrification was observed at a high DO level of 38 mg/L. However, the bacteria

Table 2-13: Range of $K_{O,A}$ and b_A reported in the literature

Specie	$K_{O,A}$ (mg/L)	b_A (1/d)	Culture	Reference
<i>Nitrosomonas</i>	0.3		pure	Loveless and Painter, 1968
	1.32 [^]		pure	Lumbers <i>et al.</i> , 1989
	0.25		pure	Peeters <i>et al.</i> , 1969
	0.32		pure	Hanaki <i>et al.</i> , 1990
		0.12 [†]	pure	
		0.039 [‡]	pure	
		0.119	AS [*]	Gee <i>et al.</i> , 1990a
		0.110	AS	Yoshioka <i>et al.</i> , 1982
		0.101	AS	Gee <i>et al.</i> , 1990b
		0.0038	AS	Rozich and Castens, 1986
<i>Nitrobacter</i>	1.84		pure	Peeters <i>et al.</i> , 1969

[^] Overall nitrification; [†] Sufficient DO; [‡] DO = 0.5 mg/L; ^{*} Activated sludge

acclimated itself to the high DO concentration after approximately 10 days. The initial high DO inhibition, however, is not a problem in the conventional activated sludge units because the DO concentration rarely exceeds 5.0 mg/L during the normal operations of the plant.

Although 1 mg/L DO is sometimes sufficient, the DO level should be maintained greater than 2 mg/L for a safe practice of operations (WPCF, 1983). But each plant has to find its own critical DO concentration which provides an optimum nitrification rate in the activated sludge units.

Effects of organic loading. The organic loading is reported to have no direct effect on the pure nitrifying bacteria cultures (Wild *et al.*, 1971; Hanaki *et al.*, 1990). However, the organic loading has an indirect effect on the nitrification process where the nitrifying bacteria is present with the other bacteria cultures such as heterotrophic bacteria.

First, high organic loading means higher BOD₅/TKN ratio and a lower nitrifier fraction in the system (see Table 2-11). Because the high concentrations of organic compounds favor the heterotrophic bacteria, the nitrifier fraction becomes an insignificant part of the biomass population. A lower number of nitrifying bacteria results in a reduction in the nitrification rate. Hall and Murphy (1985) established a correlation between COD, TKN, MCRT, and the specific substrate removal rate. They reported that when the heterotrophic bacteria was present in the system, the oxidation of ammonia and nitrite decreased with the increasing COD/TKN ratio (higher organic loading).

Second, because there is an excessive source of food available for the heterotrophic bacteria, the mass of the carbon consuming bacteria increases in the activated sludge unit. The excess portion of the bacterial mass needs to be disposed of to maintain an optimum MLSS concentration in the aeration basins. Consequently, the wasting rate is increased and along with the heterotrophic bacteria, large portions of the nitrifiers are wasted.

An optimum range of 0.3 to 0.4 lb BOD₅/lb MLVSS·d organic loading should be maintained to ensure the proper operations of the nitrification in the activated sludge processes (Balakrishnan and Eckenfelder, 1969). To adjust the organic loading, the sludge wasting rate is decreased accordingly so that there will be sufficient amounts of microorganisms available to consume the incoming high organic loading.

Effects of hydraulic retention time. The hydraulic retention time affects the nitrification process indirectly. Time for the bacteria to consume nitrogen is a critical parameter since the nitrifying bacteria fraction is normally in limited quantities. The required range of hydraulic retention time for a single-stage nitrification process is shown below (Metcalf and Eddy, 1991):

$$\theta_H = 6 \text{ to } 15 \text{ hrs} \quad (2-31)$$

where θ_H = Hydraulic retention time, hrs

The hydraulic retention time reaches critical values when there is a hydraulic overload in the plant. To achieve the desired hydraulic retention

time, diversion of the part of the influent flow into the temporary storage tanks may be required. An optimum hydraulic retention time range for the plant should be maintained to ensure the success in the nitrification process.

Effects of inhibitory compounds. There are number of inhibitory compounds that may suppress or terminate the growth of the nitrifiers (see Table 2-7). Although compounds such as phenol, cyanide, and nickel have inhibitory effects on the nitrification process, in this dissertation, ammonia concentration is considered the only inhibitory compound. Ammonia is shown to have inhibitory effects on the range of 1.0 to 10 mg/L (Anthonisen *et al.*, 1976). Since the selected refinery wastewater contains approximately 100-150 mg NH₄⁺-N/L in the influent, the effects of the high ammonia need to be considered in the nitrification kinetics.

Haldane's modification of the Monod equation can be used to represent nitrification kinetics (Haldane, 1965). The Haldane's modification introduces a new constant, the *inhibition constant* (K_i) for the Monod equation as shown below:

$$\mu_A = \mu_{m,A} \frac{S_{NH}}{K_{NH} + S_{NH} + \frac{S_{NH}^2}{K_i}} \quad (2-32)$$

where K_i = Inhibition constant, mg/L

Rozich and Castens (1986) applied the Haldane's equation and obtained a $K_i = 19.7$ mg/L while $K_{NH} = 2.7$ mg/L; $\mu_{m,A} = 1.20$ d⁻¹; and $S_{NH} = 100$ mg NH₄⁺-N/L. A similar set of experiments were performed by Gee *et al.* (1990b) to investigate the inhibitory effect of the high ammonia concentration on the ammonia utilization rate. For their model, K_i was 9,000 mg/L for *Nitrosomonas* while the maximum substrate utilization rate was 1.87 mg NH₄⁺-N/mg VSS·d; K_{NH} was 0.7 mg/L; and the yield coefficient was 0.30 mg VSS/mg NH₄⁺-N. In the same study, for *Nitrobacter*, K_i was found to be substantially lower (173 mg NO₃⁻-N/L).

Reporting a wide range of K_i values in the literature may be due to the site-specific characteristics of the K_i parameter. Particularly in the industrial wastewater treatment, the inhibition constant varies significantly based on the source and the presence of other toxic chemicals. Each treatment plant should conduct experiments to determine the K_i value for their own plant. Furthermore, the inhibition of nitrification due to high ammonia concentration may also vary depending on the ability to acclimate to the high ammonia concentration by the nitrifying bacteria, temperature, and the amount of nitrifiers in the system (Anthonisen *et al.*, 1976).

There are abundant factors which effect the nitrification process. The full understanding of these factors have been achieved not yet. Within the scope of this dissertation, the effects that are considered for the mathematical model formulation of the nitrification system kinetics are as follows: the effects of the dissolved oxygen concentration, pH, temperature, and high ammonia loading. Furthermore the fraction of the

nitrifiers, MCRT, the hydraulic retention time, and the highest allowed concentrations for the number of toxic compounds are also considered (see section 3.1.5.2).

2.1.3 Problems with Activated Sludge Process

The study conducted in 1980 showed an alarming indication of environmental hazard due to extensive number of violations by the wastewater treatment plants (General Accounting Office, 1980). In the study 242 randomly selected wastewater treatment plants were surveyed and it was found that 87% of the selected plants violated their NPDES permits.

A recent study gave more alarming indications. The survey by EPA showed 30 to 50% of the treatment plants in the U.S. are in violation of their discharge permits (Junkins *et al.*, 1983). The major reasons for failing to meet the standards are listed as: poor operations, inadequate design, unsatisfactory maintenance schedule, equipment malfunctions, uncertainties dealing with the biological systems, management failure, and absence of well established facts on the activated sludge kinetics. The study showed that although all the above mentioned factors affect the activated sludge process, poor operations is the main cause of violation. Operational problems are discussed in section 2.1.3.2 in detail.

2.1.3.1 Process Monitoring

The activated sludge process needs to be monitored 24-hours a day to make sure the environmental conditions for the microorganisms are maintained satisfactorily. Although there are several reports in the literature indicating that automated equipment in activated sludge plants is not a cost-effective alternative, it is a good practice to keep the process under constant monitoring using automated monitoring equipment. Frequent sampling and manual lab work may become frustrating after a long time.

Today there are microcomputers which can handle the data collection and monitoring efficiently with considerably less money invested (Manning and Dobs, 1980; Ostgaard *et al.*, 1985). The automated monitoring equipment may significantly reduce the possibility of violation of the standards. One example of the activated sludge process control monitoring is given in Figure 2-4. In Figure 2-4 the influent, effluent, and activated sludge parameters are monitored and displayed in the control panel located in the observation room. The operator reviews the plant conditions and may change controls to the equipment.

Sampling in the treatment plant differs with the type of variable being measured. The frequencies of sampling for different variables are extensively listed in the literature (WPCF, 1980). In practice, the samples are collected as a *grab sample* (*i.e.*, exhibits the characteristics of the wastewater at the time the sample is taken).

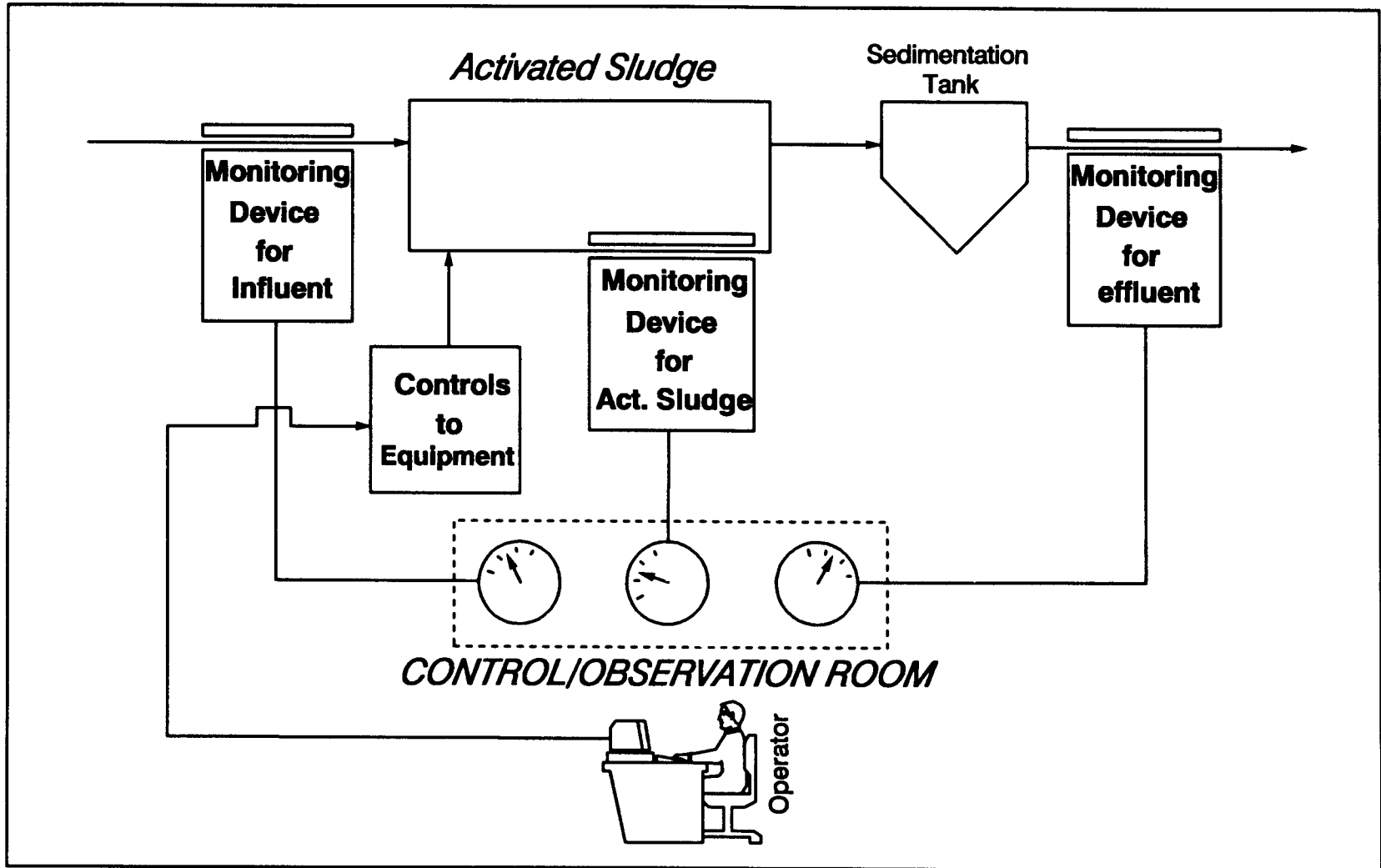


Figure 2-4: Monitoring of the activated sludge process

A more representative sampling method, *composite sampling*, combines the grab sample values and takes an average over a period of time. The most frequent time period is 24-hour. The composite sampling is used because it gives the overall characteristics of wastewater by eliminating the possible irregular fluctuations in the wastewater characteristics and the errors made in the lab procedures.

The sampling can be performed either manually or automatically. Manual sampling most often requires time consuming lab work. Automated instrumentation, however, uses probes, optical devices, and mechanical parts to measure the variables in less time. Unfortunately, not all the necessary tests can be performed by automatic equipment at a reasonable cost.

Many treatment plants today have both manual and automated sampling systems. The variables needed to monitor in the activated sludge process are listed in Table 2-14.

2.1.3.2 Poor Operations

The most frequent cause of violation of the NPDES permit is reported as unsatisfactory operations of the wastewater treatment plant (Junkins *et al.*, 1983). Qualifications necessary to become an operator are minimal. In the State of California, there are basically four types of job categories available for the treatment plant operators (see Figure 2-5) (Ludwig, 1989).

Table 2-14: Activated sludge process sampling locations, analysis and frequencies (WPCF, WASTEWATER SAMPLING FOR PROCESS AND QUALITY CONTROL, ©1980. p. 8. Reprinted by permission of WPCF, Alexandria, VA)

Sampling Location	Analysis	Sample		
		Use	Frequency	Type
Primary effluent	BOD	PP [†]	D ^{†◊}	C [◊]
	TSS	PP	D [†]	C
	pH	PC [^]	D [†]	G [*]
Mixed liquor	DO	PC	D	G
	Temperature	PC	D	G
	TSS	PC	D [†]	C
	VSS	PC	D [†]	C
	NO ₃	PC	1/wk	G
Return sludge	TSS	PC	D [†]	C
Final clarifier effluent	BOD	PP	D [†]	C
	TSS	PP	D [†]	C
	DO	PP	D	G
	Fecal Coliforms	PP	1/wk	G
	Cl ₂ residual	PP	D	G
	pH	PP	D	G
	TKN	PP	1/wk	G
	NH ₃	PP	1/wk	G
	NO ₂	PP	1/wk	G
NO ₃	PP	1/wk	G	

[†]Frequency could be reduced if the discharged permit allowed and/or process upsets were infrequent.

[†]PP: Plant performance

[^]PC: Process control

[◊]D: Daily

[◊]C: Composite sample

^{*}G: Grab sample

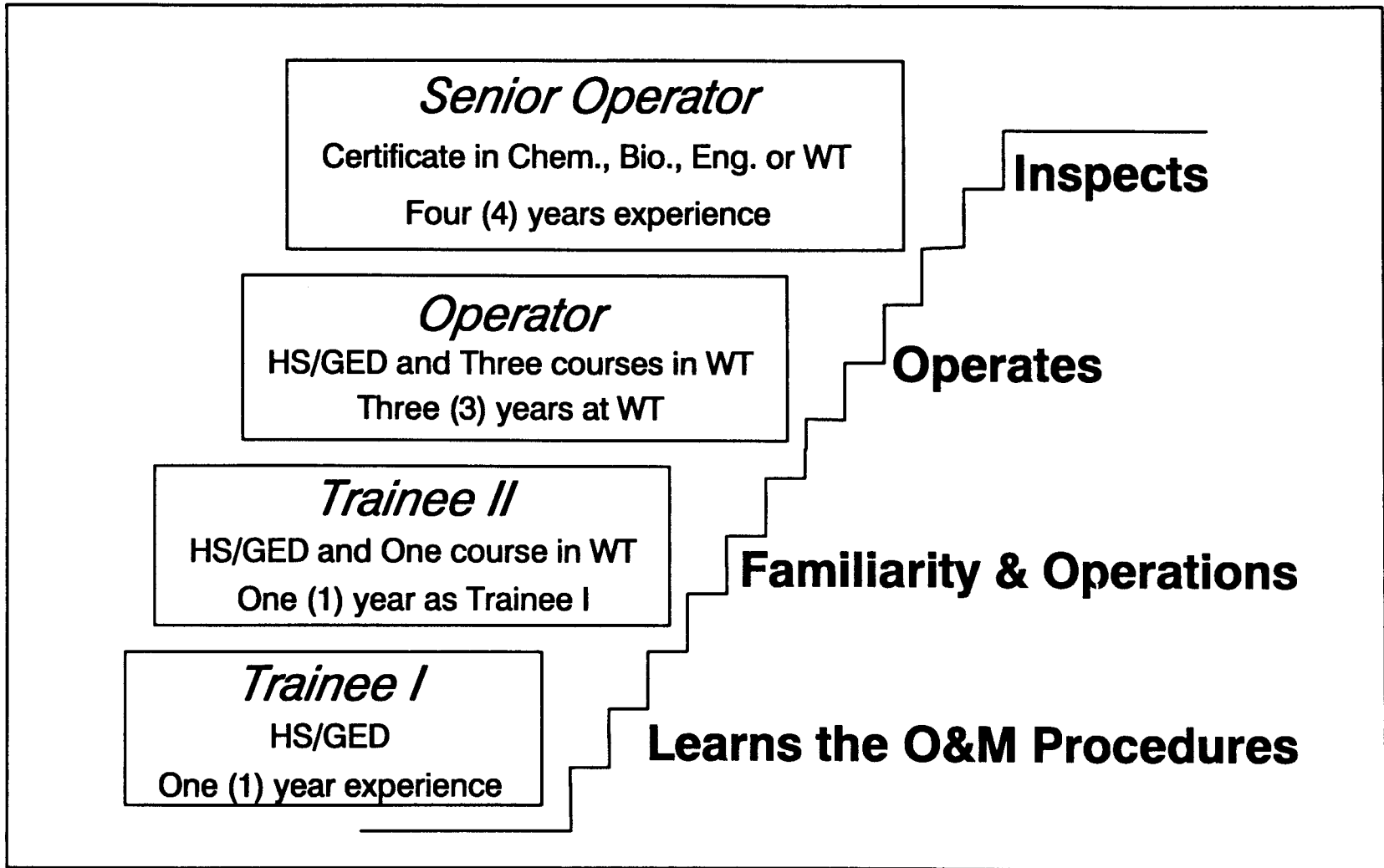


Figure 2-5: Career ladder for wastewater operators in the State of California

At the very bottom of the ladder is the *Trainee I* operator. Education requirements include only a high school diploma (HS) or general education degree (GED) and a year of mechanical work experience. The Trainee I's job is to learn operations and maintenance procedures. At this stage the Trainee I operator is not allowed to operate the plant without supervision.

After a year in the Trainee I position and an additional course in water treatment, a person can become a *Trainee II* operator. In this step of the career ladder, the operator is given the responsibility of operating the system while he/she becomes familiar with the real-time plant operations.

Within a three year period a Trainee I can advance to the level of *Operator* with the only requirement being the completion of three courses in the water treatment field. As an operator, he/she has complete responsibility for the operation and maintenance of the plant.

The top of the career ladder is the *Senior Operator* who inspects all other operators. This position requires a certification in chemistry, biology, or engineering with four years experience in the water treatment field.

These job categories are filled by completion of state training courses. The state offers a correspondence course available to anyone with a high school degree which allows them to become a Trainee I in as little as 15 weeks. The major problem is that activated sludge process is a 24-hour system. Because high ranking operators are scheduled for preferred daytime shifts, the night shifts frequently are staffed solely by

unsupervised, inexperienced Trainee I & II level operators. These rookie operators have not obtained sufficient practical experience nor received adequate information to handle potential problems. Consequently, they often make mistakes and cause serious wastewater spills to the environment. Late night shifts frequently are staffed by the lowest ranked operators and many problems result during this time. It is essential to demand tougher course requirements to qualify as an operator. Major problems are related to the following issues:

- ☛ Inadequate requirements on operators' curriculum
- ☛ Heavy responsibilities given to inexperienced trainee operators
 - ✓ Insufficient knowledge of activated sludge problems
 - ✓ Lack of practical experience
 - ✓ Inaccurate decision making ability

The above listed problems may be eliminated by introducing extensive operator training programs. The benefits obtained from such programs are invaluable with respect to the enhancement achieved in the completion of the training courses. The program may quantitatively and qualitatively be evaluated to identify the benefits acquired (Kerri, 1991).

2.1.3.3 Nitrification Failure

Nitrification in the activated sludge process is an extremely sensitive process due to the presence of numerous affecting factors. A delicate

balance between the required parameters must be obtained for the proper operations of the nitrification process. Parameters that have significant effect on the nitrification process include; MCRT, pH, temperature, dissolved oxygen concentration, organic loading, and hydraulic retention time. Moreover, adequate amounts of required nutrients for the nitrifiers and the absence of toxic chemicals must be satisfied to achieve problem-free nitrification process operations. The individual effects of these factors on the nitrification process are discussed in section 2.1.2.3.

MCRT is a key parameter in operating the nitrification process. Therefore, great attention should be given to regulate the MCRT fluctuations by adopting an accurate process control method for the MCRT control. Based on the studies conducted in real-time activated sludge processes, it is recommended that in order to achieve a complete nitrification, MCRT must be equal to or greater than 20 days (WPCF, 1987). Lesser MCRT values indicate a problem with the nitrification process. MCRT should be kept around 20 days by adjusting the sludge wasting rate (see section 2.1.4.3).

The pH is another important factor in the nitrification process. Effect of pH is significant on the growth kinetics of the nitrifying bacteria. A proper pH range of 7.0 to 8.0 should be maintained always for the optimum activity of the nitrifiers. Because raising pH from 7.0 to 8.0 does not constitute a significant nitrification rate increase (approximately less than 10% change in the rate), the optimum pH for the real-time operations should be approximately 7.2. Furthermore, increasing pH to

a higher value requires an additional chemical cost. For the proper operations of the nitrification process, pH must be controlled to avoid problems with the process.

Temperature has an additional effect on the nitrification process kinetics. The growth rate of the nitrifying bacteria is closely correlated with the optimum temperature range. For an optimum nitrifier growth rate, temperature should be between 30 to 35°C. If the temperature drops below 18°C or raises above 35°C, the growth rate of nitrifiers decreases significantly. In some cases where the temperature is below 10°C, the nitrification process is considered completely suppressed. The temperature effect becomes more significant during the cold days of winter. Although many treatment plants do not have much control over the temperature in the aeration basins, the source which is causing the temperature decrease should be eliminated to achieve a complete nitrification in the activated sludge process.

The dissolved oxygen concentration is a requirement in the nitrification process. The nitrifying bacteria is obligate aerobe and depends on the oxygen to oxidize the ammonia and nitrite. If DO drops below the limits, a significant reduction in the nitrification rate is observed. In the real-time activated sludge process operations, DO should be kept above 2 mg/L to eliminate the substrate limiting factor of DO and to provide an aerobic environment for the nitrifying bacteria to thrive. Constant process control of DO is required to ensure the completion of nitrification process. If DO drops below 1 mg/L level, additional blowers should be put in service to

raise the DO concentration before the nitrification process is damaged. Although it depends on the other environmental conditions, generally the DO levels below 0.5 mg/L indicate the complete inhibition of the nitrification process in the real-time activated sludge process operations.

Organic loading may also create problems with the nitrification process. Although high organic loading does not have a direct effect on the nitrifying bacteria growth or activity because it results in the domination of the heterotrophic bacteria within the biomass population, high organic loading indirectly causes nitrification failure. Organic loading should never be allowed to exceed 0.5 lb BOD₅/lb MLVSS·d levels to ensure the proper operations of the nitrification process.

Low hydraulic retention time may cause additional problems with the nitrification process. If there is not enough time for the nitrifying bacteria to reside in the aeration basins, a reduction in the nitrification rate is observed. Low hydraulic retention time is caused by increased influent flowrate. Thus, diverting the part of the influent flowrate would increase the hydraulic retention time in the activated sludge units.

Finally, absence of adequate amounts of required nutrients for the nitrifiers and presence of toxic chemicals may also cause nitrification to fail. Sufficient amounts of nutrients must be available to the nitrifiers for their growth and activities. Some of the nutrients may have inhibitory effects on the nitrifying bacteria at high concentrations. Inhibitory effects are discussed in section 2.1.2.1. In addition to inhibitors, there might be

chemicals in the water which are toxic to the bacteria. Particularly in the refinery wastewater, there are potentially more toxic chemicals which may cause a complete termination of the nitrifying bacteria growth and activities. The potential toxics are given in Table 2-7.

As discussed in this section, nitrification is a delicate process in which all the required conditions must be satisfied for the optimum performance. Any disturbances in the parameters which affect the nitrification result in reduction of the nitrification rate. If these disturbances are in large magnitudes, the nitrification process is significantly reduced or even completely terminated.

2.1.3.4 Bulking

One of the problems frequently faced by wastewater operators is *sludge bulking*. Bulking sludge is sludge which has lost its ability to settle properly. The survey conducted on the US activated sludge treatment plants showed that approximately 80% of the plants had a bulking incident at least once during a one year period of time. The sludge bulking problem is world-wide. Approximately 63% of the plants surveyed in the UK experienced bulking problems (Tomlinson, 1982). Similar results were obtained from the study in Germany. The results indicated that approximately 70% of the plants observed SVI values of over 150 ml/g for a time period of two months in a year (Wagner, 1982).

Because of the magnitude of the problem, many researchers worked on the bulking problem. The authors who worked on the causes of the bulking problem discovered several aspects of this problem including presence of filamentous organisms (Pasveer, 1969; Farquhar and Boyle, 1971; Van den Eynde *et al.*, 1982). Closer investigation of the filamentous organisms showed that these organisms are the major cause of the bulking (Eikelboom, 1975; Green, 1982; Slijkhuis and Deinema, 1982). The studies of the kinetics of the microorganisms showed that filamentous organisms have lower Monod constants than the floc-formers' (Van den Eynde *et al.*, 1982; Chudoba *et al.*, 1985; Richard *et al.*, 1985).

On the control and prevention of bulking, authors focused on microbiological characteristics (Chudoba *et al.*, 1985), plant's physical characteristics (Chudoba *et al.*, 1973b; Rensink, 1974; Chambers, 1982; Beck, 1984), operational parameters and the possible control strategies (Tenney and Stumm, 1965; Chudoba *et al.*, 1973a; Finger, 1973; Chudoba *et al.*, 1974; Chapman *et al.*, 1976; Palm *et al.*, 1980; Eikelboom, 1982; Jenkins *et al.*, 1982; Rachwal *et al.*, 1982; Rensink *et al.*, 1982; Waller and Hurley, 1982; Wilson, 1983; Daigger *et al.* 1985; Geselbracht *et al.*, 1988; Van Niekerk *et al.*, 1988).

In the normal operation of the activated sludge plants, the bacteria is shaped in a *floc* form. When the bulking occurs, the floc formation is altered and the specific gravity of the floc is reduced (Figure 2-6). Lower specific gravity results in poor settlement quality of the sludge. The reduction of the specific gravity is accomplished basically in two ways:

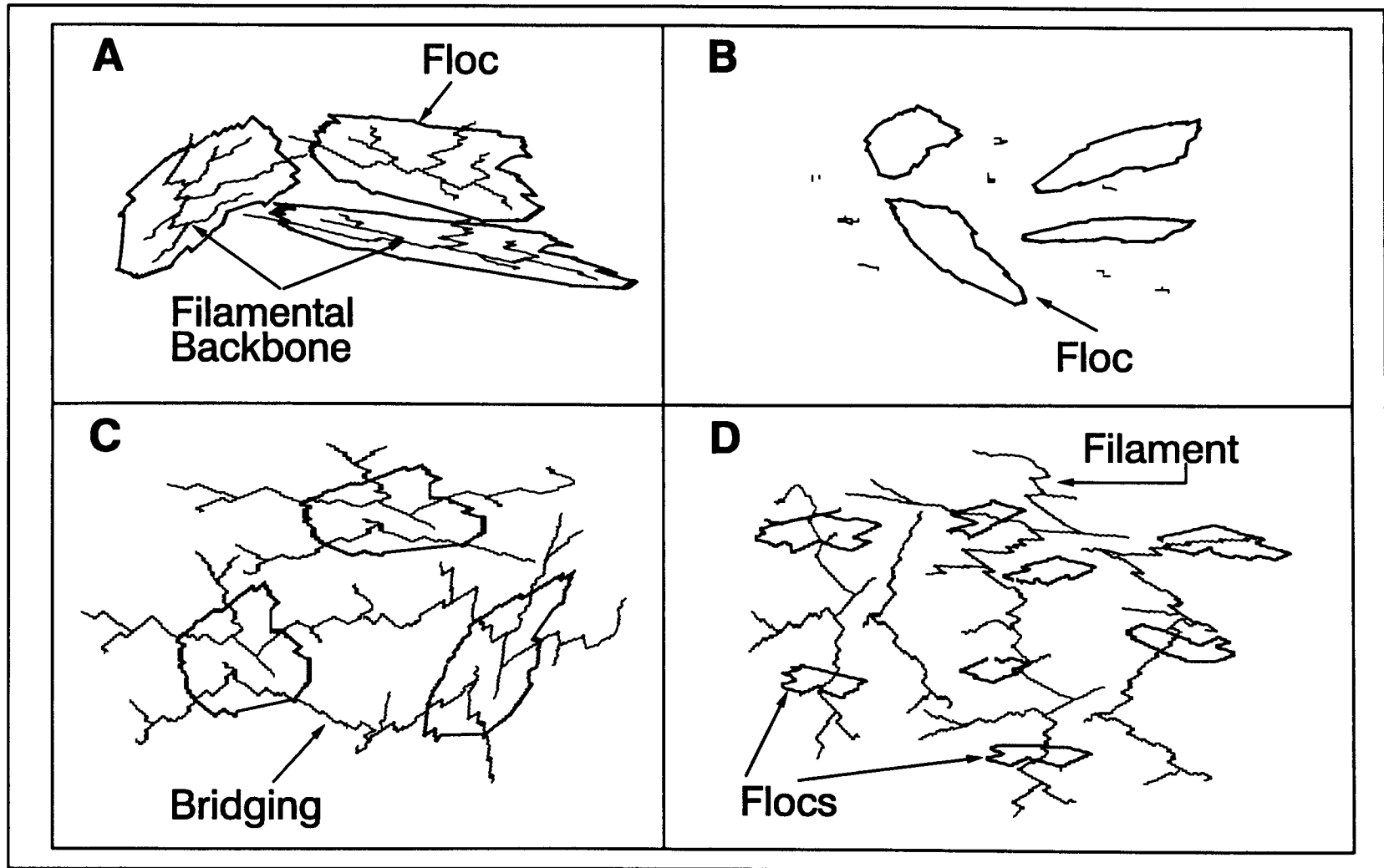


Figure 2-6: (a)Normal condition; (b)No filaments present; (c)Filament bridging (Type I bulking); (d)Presents of few flocs (Type II bulking)

- 1) Filamentous organism bridge between flocs and keep these flocs apart (Type I bulking: see Figure 2-6c).
- 2) Floc-formers don't have a chance to get together because filamentous organisms grow within the floc and cause porous floc structure (Type II bulking: see Figure 2-6d).

Either way the floc loses its settling and compaction abilities. When the filamentous organisms keep the already developed flocs apart, the ability of compaction is dramatically reduced. The sludge will not settle or compact properly. The other type of effect of filamentous organisms basically causes the same result. While the flocs are forming, the filamentous organisms grow inside the floc and create a grainy floc formation. In this case internal fluid concentration of the floc is increased in amounts that prevent the floc from settling effectively. The causes of sludge bulking and the filamentous organisms involved in bulking are summarized in Table 2-15.

In Table 2-15, the probable causes of filamentous bulking are identified as follows: filamentous bulking may be caused by low DO (less than 0.5 mg/L), insufficient nutrients present, low F/M (0.05 - 0.2 lb BOD₅/lb MLVSS·d), or by low pH (less than 6.5). In the bulking episodes which no filamentous organisms are involved, the probable causes for the bulking problem include too high or too low F/M, excessive turbulence in aeration basins, and the presence of toxic chemicals in the influent.

Table 2-15: The probable causes of filamentous bulking are suggested as (compiled from WPCF, 1987 and Richard, 1989):

The probable causes of filamentous bulking	Filamentous organisms involved
Low dissolved oxygen concentration in activated sludge basins	Type 1701, <i>Sphaerotilus</i> spp., <i>Haliscomenobacter hydrossis</i>
Nitrogen and/or Phosphate deficiencies	<i>Thiothrix</i> spp., Type 021N, Type 0041
Low F/M	<i>Microthrix parvicella</i> , <i>H. hydrossis</i> , Type 021N, Type 0041
Low pH	Fungi
High sulfide concentration (septic waste)	<i>Thiothrix</i> spp., Type 021N, <i>Beggiatoa</i> spp.
Too high or too low F/M	No filamentous organisms involved
Excessive turbulence	
Toxic chemicals in the influent	

2.1.3.5 Other Settlement Problems

Besides sludge bulking, the activated sludge has other sludge settlement problems. Major sludge problems are briefly defined and corrective actions summarized in this section.

Floating sludge. When the sludge rises to the clarifier surface, it is referred to as floating sludge. The cause of rising sludge is identified as denitrification in the sludge blanket in the clarifier tank. The denitrification process produces nitrogen gas and the gas released from the

bottom of the clarifier starts rising to the top. While the gas bubble is going up, it catches the flocs which are settling and carries them to the surface. To correct the problem, the sludge removal rate from the clarifier should be increased. The sludge should not be allowed to stay in the clarifier in a prolonged time to cause denitrification. If possible DO in the aeration basin should be raised. To reduce the retention time of sludge in the sedimentation tanks, reduction in the amount of sedimentation tanks on line may be necessary to move the sludge out more frequently.

Pinpoint floc. One of the more common problems is when the sludge does not form good flocs. When there is overaeration in the basins, floc shearing may occur. Broken flocs have less ability to settle. The same effect is observed when the activated sludge is operated in the long sludge ages. A good mixture of filamentous and floc-former organisms should be present to achieve a well-formed floc. If the necessary amount of filamentous organisms are missing, the floc-formers lose their backbone and the flocs become individual small flocs which have a limited ability to settle. Corrective measures such as reduction of turbulence in the aeration basins and reduced sludge age operation may prevent the problem of pinpoint floc.

Ashing. When very small sized sludge floats on the surface of the clarifier, it is called ashing. The phenomena is probably caused by denitrification in the clarifier, low F/M, or when grease concentration is high. Raising the F/M ratio and setting the correct MCRT can solve part

of the problem. If grease is the causative factor, the grease needs to be removed by scum collection systems.

Straggler floc. When sludge appears as small, fluffy flocs, it is called straggler floc. The light flocs rise to the surface of the clarifier sometimes creating a turbid effluent. If this is the case, the problem is probably caused by low MCRT (*i.e.*, high F/M ratio). The sludge wasting should be decreased to raise the MCRT and the MLSS concentration in the aeration basins.

2.1.3.6 Foaming

Foaming is one of the most common activated sludge problems. The microorganisms responsible for foaming are: *Nocardia* spp., *Microthrix parvicella*, and rarely Type 1863 (Richard, 1989). Although the latter two are causes of foaming, *Nocardia* spp. is the most common microorganism accountable for foaming problems in the activated sludge process. The foaming problem is observed in approximately 40% of the treatment plants at least once during the year. Even though *Nocardia* spp. is a filamentous organism, it is not the cause of bulking due to its short filamentous structure. If there is a foaming problem, it can occur either in the aeration basins or on the clarifier surface.

In some severe cases of foaming, the foam may spill over the walkways and endanger the operator's personal safety. Under normal operations in

the activated sludge process, approximately 25% of the tank surface is covered with light brown foam. The normal thickness of the foam should be 2 to 3 inch. There are basically three different kinds of foaming problems in the activated sludge process.

White foam. Stiff white foam is an indication of "young sludge" or low MCRT. Although the appearance of small amounts of white foam are acceptable for effective operation, an excessive amount indicates there is not enough MLSS in the tank and the F/M ratio is too high. Corrective measures may be taken such as stopping sludge wasting until adequate MLSS is compiled in the aeration basins. The isolation of the foam from the surface and feeding it back to the aeration basin slowly in the concentrations that will not alter the normal activated sludge operations may eliminate the white foam problems.

Thick brown foam. If excessive brown foam occurs in the plant, it is probably because of low loading rates. Low F/M means there are excessive amounts of MLSS in the aeration tanks. This might have been caused by the warmer temperatures favoring the optimum growth of the microorganisms or decreased sludge wasting. To ensure the normal operations of the activated sludge process, sludge wasting should be increased to raise the F/M ratio. Decreasing MLSS, however, would interfere with the nitrification process. If the treatment plant has to nitrify, then increasing sludge wasting is not appropriate. Other physical solutions such as surface water spraying should be considered.

Black foam. Dark or black foam is evidence of insufficient aeration. The anaerobic conditions favor the growth of foam causing organisms and results in black foam accumulation. As corrective actions, the aeration rate should be increased and complete mixing in the basins should be obtained. If there is enough organic removal already existing, lowering the MLSS concentration may also be possible. Sometimes, the dark color of foam is due to industrial dye spills. In this case the source of the dye spills should be eliminated.

In all cases of foaming problems, short term control strategies such as surface water spraying, adjustments of the MCRT and F/M ratio, or adding anti-foaming agents is commonly practiced. For the long term solutions, control of grease in the influent would decrease the foaming problems.

2.1.3.7 Minor Causes of Activated Sludge Problems

The design of the treatment plant plays an important role in the operations. Inadequate design may cause serious problems. Design problems often are very costly to overcome. Limitations in the equipment, such as an inadequate maximum capacity of a pump, may cause problems. In addition to the equipment limitations, the flexibility of the design is an important factor to achieve an optimally efficient operation. The operator should be able to try different modifications of the procedure without having to go through major adjustments.

The equipment operations should be easy to understand and easy to control. Complicated devices do not always mean better devices. The equipment which comes with a complex, hieroglyphic-like, thick manual is more likely to cause a failure.

The characteristics of wastewater never reach a steady state value. The different values of incoming wastewater makes things difficult to control. Each day presents different challenges in the wastewater treatment plant. The unknown is always present. Fluctuations in the characteristics should be monitored and data patterns, if possible, should be developed.

The uncertainties of the biological systems make things more difficult. No one yet understands the biological systems so well that we can predict the future. All the studies are limited to simulating the system to a degree. The unknown in the biological world brings unexpected responses from the microorganisms. One day the process control strategy may work but the next day it may not.

During equipment malfunctions, there are few alternatives that may be exercised. If the equipment is not supported by a spare, the operator has to bypass the process and discharge the wastewater with limited treatment. The equipment malfunctions are often caused by insufficient maintenance of the equipment. Strict maintenance schedules should be followed to eliminate the problems with break-down. The equipment replacement schedules are also important to follow. The life expectancy of equipment in wastewater treatment plants is generally much lower than in other fields.

The suspended solids concentration of the wastewater dramatically lowers the life of the pumps, lines, connections, valves, gates, and sampling apparatus. Frequent cleaning of vital equipment, such as influent pumps, valves, air pumps, and sludge pumps, would prevent unnecessary wastewater spills due to equipment malfunction.

Other problems are mainly due to the aeration basin problems. In the diffused aeration systems, there could be several problems, such as, "dead spots" due to inadequate mixing, violent turbulence due to overaeration, or clogged diffusers. The DO throughout the aeration basin should be between 1.5 - 4.0 mg/L. If this DO range is not supplied, there might be anoxic or anaerobic conditions in the aeration basin. The problem is probably caused by improper distribution of air, clogged diffusers, or misplacement of diffusers in the basin. To correct the problem, clean diffusers, adjust air flow for different sections of the tank so that constant DO concentration will be achieved throughout the basin, or, if possible, reorganize the diffusers in the basin.

Similar problems exist for mechanical aeration systems. Inadequate mixing patterns in the basins are caused by insufficient aerator speed or an improperly designed impeller. If the problems are caused by mechanical equipment failure, the management should replace or repair the equipment without delay.

2.1.4 Process Control and Troubleshooting

In this section, the process control strategies and troubleshooting will be discussed. Under normal operating conditions none of the problems given in the previous section should exist. The aim of all the operators should be "prevention" of the problems. Theoretically, optimum operation of an activated sludge process can be achieved. In practice, however, the operators face many serious problems every day.

When a problem occurs, the operator should, along with temporary corrective actions, also take long term preventive actions. In the previous sections, the major problems with the activated sludge process and possible corrective actions are discussed. Now this dissertation will focus on the most important troubleshooting strategies to prevent nitrification failure, bulking, foaming, and other settlement problems.

In the aeration basins, control of air, maintenance of adequate MLSS concentration, chemical additions, and nutrient balance are some of the most important control strategies. In the clarifiers, sludge handling is the most important control strategy.

2.1.4.1 Air Control

The dissolved oxygen concentration is a vital process control parameter in the activated sludge process. The amount of oxygen available for the microorganisms is virtually the most important aspect of their survival.

The optimum DO of 1.5 to 4 mg/L should always be maintained to ensure system stability. The DO in the aeration basins is a function of both hydraulic and organic loading. Since neither loading parameters are stable in the treatment plant, the DO demand in the basins may never be constant. Based on his/her experience, the operator will know how much air to supply to the aeration basins by observing the flowrate and the incoming organic loading. The experienced operator would keep the DO levels in a satisfactory range.

In case of mechanical failure, if the DO drops below 0.5 mg/L, the biomass may incur serious damage. Constant monitoring of the DO throughout the aeration basin is necessary. If DO is low, the stand-by air blowers should be put on line and frequent cleaning of the diffusers should be performed. If the DO concentration is excessive, in addition to wasted energy, there will be breakage of flocs and the occurrence of settlement problems.

The plant operator controls the DO concentration by either adjusting the speed of the air pumps or by throttling the air valves. Typically, a butterfly valve and a venturi tube are placed where the air blower intakes the air (see Figure 2-7).

The operator reads the air flow measured by the venturi tube and then adjusts the butterfly valve depending on the amount of air desired. These two apparatus can also be placed after the air blower and before the diffusers. The DO is measured constantly by several probes in the

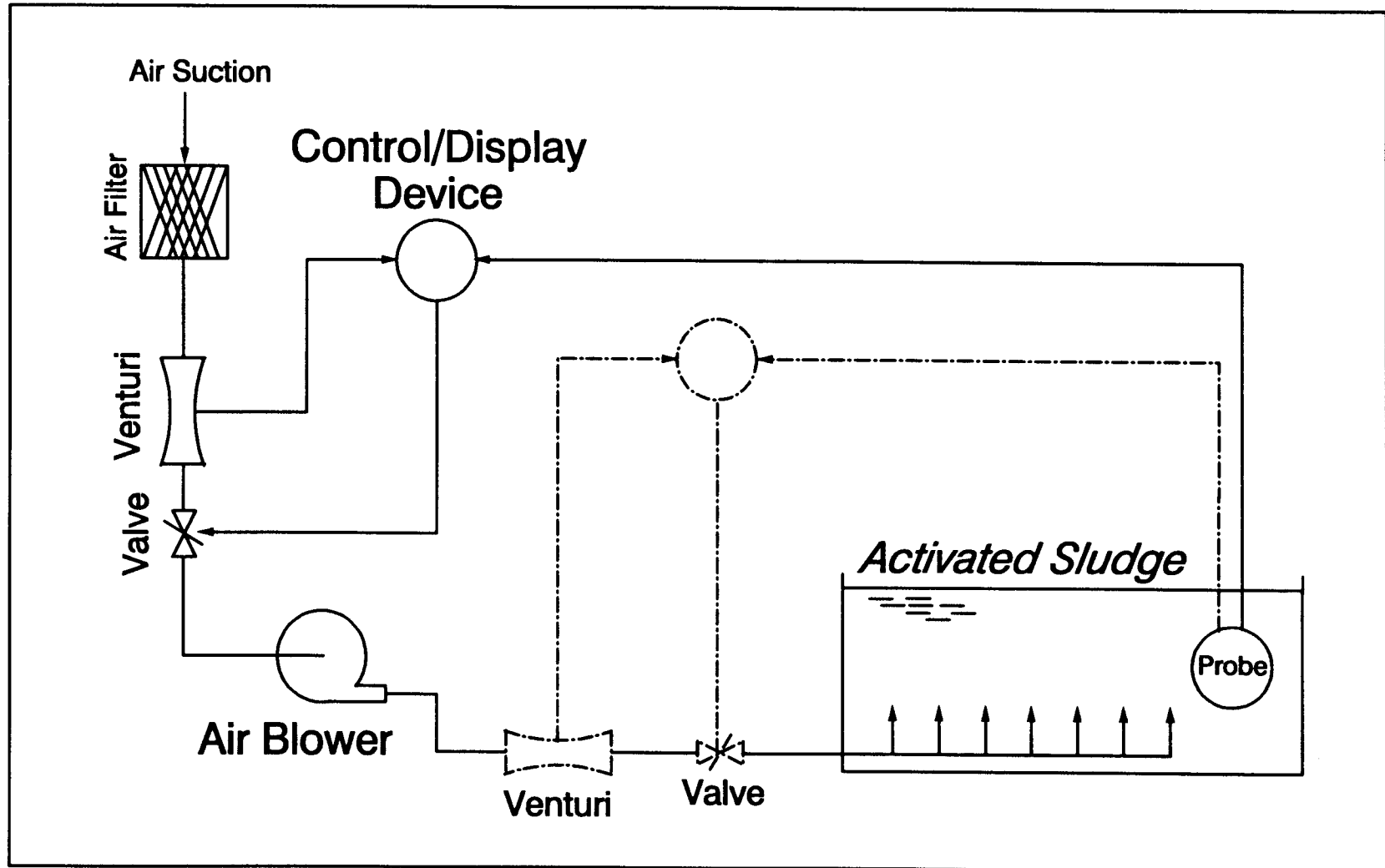


Figure 2-7: Control of dissolved oxygen concentration in activated sludge process

aeration basins so that the operator can continuously observe DO from the control room. When any changes are made, there is a lag time until the operator observes a constant DO. Most of the time the operators overshoot the DO and try to adjust it by trial and error. The constant values of DO are normally very difficult to achieve due to unsteady state characteristics of the incoming wastewater.

In the basic manual control, all the equipment is controlled by the operator and reliability of the future outcome actually depends on how much experience the operator has. Computerized control of air blowers is also possible. Preset values of upper and lower limits for the DO help the computer to adjust the air rate and supply adequate air to the aeration basins. Although the automated process control devices are not used extensively today, tomorrow's plants will use computerized process control apparatus widely. If there is any automated equipment available with a reasonable maintenance requirement, the plant management should consider updating their system, as computer analysis of process control may be more reliable than an inexperienced operator.

2.1.4.2 Sludge Return

The amount of RAS is measured based on the flowrate (Q_r). The returned sludge typically is introduced at the beginning of the aeration basins. There is generally a wet well for the RAS just before the sludge is fed back into the aeration basins. Figure 2-8 illustrates the RAS process

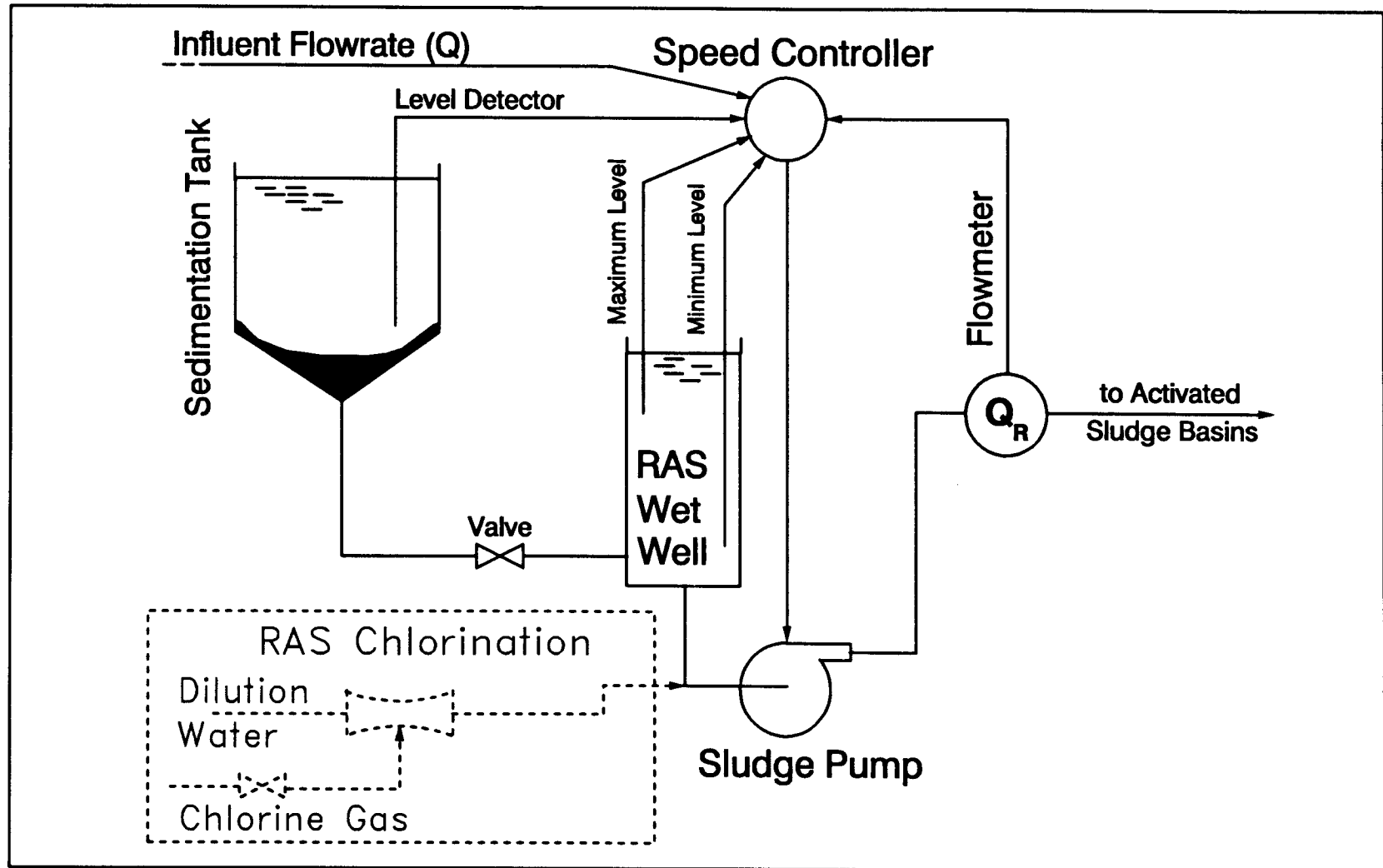


Figure 2-8: Control of the return activated sludge rate

control diagram.

The factors effecting the RAS are the influent flowrate, the aeration basin MLSS, the sludge MLSS, and the organic loading. The flowrate has a much greater effect on the RAS because higher flowrates require more biomass to treat the wastewater to the desired quality. Although other factors including the MLSS concentrations and the organic loading have a less significant effect on determining the RAS rate, the RAS rate should be adjusted based on the changes in all the parameters given above.

In practice, there are a number of techniques to control the RAS rate. In this section, five of these methods including the constant return ratio, the constant RAS flowrate, the direct sludge blanket depth control, the secondary clarifier mass balance, and the aeration basin mass balance methods are described.

Constant return ratio. One method of process control is maintaining a constant return sludge percentage. The amount of RAS is determined by the incoming wastewater flowrate (Q). The ratio (Q_r / Q) is used to ascertain the volume of RAS needed. This ratio ranges vastly and depends on process modification; however, the typical range is 25 to 100 percent. The estimation of the RAS rate by using the constant return ratio method is calculated by the following equation:

$$Q_r = \frac{C}{100} Q \quad (2-33)$$

where Q_r = Return Activated Sludge rate, gpm
 Q = Influent flowrate, gpm
 C = Constant return ratio, %

Once the ratio is determined, the operations are verified by measuring the MLSS concentration in the basins to ensure the presence of sufficient amounts of MLSS. The disadvantage of maintaining a constant Q_r / Q ratio is that frequent adjustments to the sludge pumps are required because of fluctuations in the influent flowrate. The operator is thus required to spend excessive attention on the pump operations.

Constant RAS flowrate. Another process control strategy, which is a simple method of operations, is to maintain a constant RAS flowrate. But a constant RAS flowrate will cause fluctuations in the F/M ratio and MCRT values. Because of an improper RAS rate, the desired treatment may not be reached in certain cases. Also, there is always a possibility of a sludge blanket overflow in the clarifier when the daily peak flowrate is received by the plant.

Direct sludge blanket control. A more reliable process control of the RAS rate may be achieved by the direct sludge blanket control method. With this method, the sludge blanket levels are controlled by the amount of sludge pumped out of the clarifiers. A target sludge level (1 to 3 ft) is obtained by adjusting the RAS rate accordingly.

The sludge level in the clarifiers is also considered in the operations of the wet well levels. If the sludge return rate is less than required due to the

accumulation of the excess sludge in the clarifiers, the sludge blanket raises and part of the solids escape to the effluent which result in a decreased quality of treated water. Furthermore, more attention should be paid to the sludge blanket levels particularly during high hydraulic loading periods.

Secondary clarifier mass balance. Mass balance calculations provide an additional option for the control of the RAS rate. The RAS rate can be calculated based on the solids entering and exiting the secondary clarifiers. Assuming the RAS suspended solids concentration is equal to the waste sludge suspended solids, the following equation is used to estimate the RAS rate based on the secondary clarifier mass balance calculations (WPCF, 1987):

$$Q_r = \frac{X Q - X_w Q_w}{X_w - X} \quad (2-34)$$

where X = Aeration basin MLSS, mg/L
 X_w = Waste sludge SS, mg/L
 Q_w = Waste sludge flowrate, gpm

The RAS rate should be determined on a daily basis by observing various MLSS concentrations around the sedimentation tanks. If an adjustment is needed, the RAS rate should be adjusted by not more than 10% increments of the original RAS rate.

Aeration basin mass balance. Perhaps one of the most widely used controls of the RAS rate is performed by the use of aeration basin mass

balance calculations. Similar to the sedimentation tank mass balance, the solids entering and exiting the aeration basins provide an alternative method to estimating the RAS rate. Assuming the RAS suspended solids is equal to waste sludge suspended solids, the following equation is used to determine the RAS rate based on the mass balance around the aeration basins (WPCF, 1987):

$$Q_r = \frac{X Q}{X_w - X} \quad (2-35)$$

Similar to the secondary clarifier mass balance method, the RAS rate should be determined daily by using Eq. 2-35. Adjustments needed on the RAS rate should be made slowly to eliminate unsteady conditions within the system.

The process control of the RAS rate can be achieved with the equipment shown in Figure 2-8. The operator records the influent flowrate, MLSS in RAS and in the aeration basins and makes a decision on the volume of RAS to feed back into the aeration basins by using one of the methods given above. The RAS pumps are generally centrifugal pumps so that the adjustment on the speed delivers the desired volume of sludge. The operator adjusts the speed of the RAS pumps accordingly and there is a minimal lag time until the MLSS concentration reaches its steady state value.

2.1.4.3 Sludge Wasting

Under normal conditions, there will be an excess of bacteria mass generated by the activated sludge process. This part of the biomass has to be disposed of. WAS is taken from the sludge line by centrifugal sludge pumps (see Figure 2-9). The operator can control the amount of WAS by adjusting the speed of the sludge pumps. This portion of the sludge is then taken to sludge treatment facilities for further treatment.

The amount of WAS is generally expressed by lb SS/d. The volume is then calculated for a given MLSS concentration in the sludge line. Depending on the adopted control strategy, the WAS is calculated by either obtaining a constant MCRT, a constant F/M ratio, or maintenance of a constant MLVSS concentration in the aeration basins.

Constant MCRT. Maintaining a constant MCRT is the most common process control technique for the WAS operations. MCRT is calculated by using Eq. 2-2. A constant MCRT may be maintained by adjusting the WAS rate accordingly. The following equation is used to calculate the WAS rate to maintain a constant MCRT:

$$Q_w = \frac{V X}{(C_{MCRT}) (X_w) (24) (60)} - \frac{Q X_e}{X_w} \quad (2-36)$$

where C_{MCRT} = Constant MCRT value, d

X_e = Effluent TSS concentration, mg/L

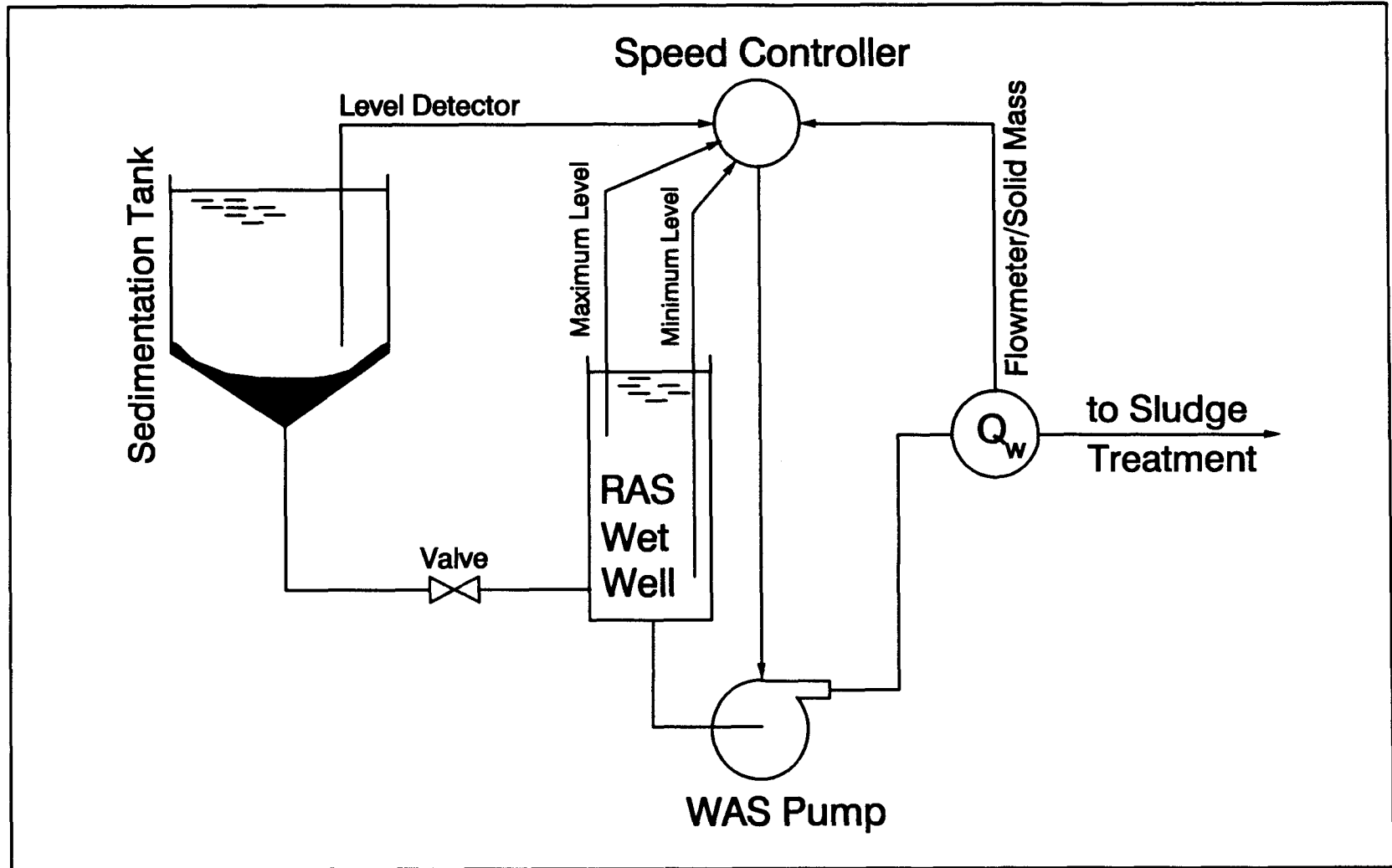


Figure 2-9: Control of the waste activated sludge rate

The constant MCRT represents the optimum value for the desired plant performance. Because the bacterial activity is affected by the temperature, the constant MCRT value should be adjusted based on seasonal variations.

Constant F/M ratio. Another method to control the WAS rate is to maintain a constant F/M ratio in the aeration basins. The sludge wasting rate is used to adjust the MLSS concentration in the aeration basins according to the incoming organic load. The F/M ratio represents the available amount of organics for the microorganism in the aeration basins per day (see Eq. 2-4). The amount to be wasted can be estimated by the following equations:

$$X'_{vss} = \frac{(Q) (S_o) (24) (60)}{V C_{F/M}} \quad (2-37)$$

$$Q_w = \frac{(X_{vss} - X'_{vss}) V}{(X_w) (P_{vss}) (24) (60)} \quad (2-38)$$

where X'_{vss} = Target MLVSS concentration, mg/L

S_o = Influent BOD₅ concentration, mg/L

$C_{F/M}$ = Constant F/M ratio, 1/d

X_{vss} = Aeration tank MLVSS concentration, mg/L

P_{vss} = Percent VSS content (MLVSS/MLSS), %

The estimation of the WAS rate by using Eq. 2-38 is not as accurate as the estimation made based on the constant MCRT value (see Eq. 2-36). If an adjustment is needed to maintain a constant F/M ratio, the waste flowrate is changed accordingly.

Constant MLVSS concentration. Maintaining a constant MLVSS concentration in the aeration basins is a simpler but not an accurate method of process control of the WAS rate. To successfully obtain a constant MLVSS concentration, the influent organic loading should be almost constant. Once the optimum MLVSS concentration is determined based on past experiences with the plant performance, the waste rate is adjusted to maintain that optimum MLVSS value in the aeration basins. The following equation is used in determining the WAS rate based on maintaining the constant MLVSS concentration:

$$Q_w = \frac{(X_{vss} - C_{vss}) V}{(X_w) (P_{vss}) (24) (60)} \quad (2-39)$$

where C_{vss} = Constant MLVSS concentration, mg/L

For each of the various control strategies mentioned above, there will be a different estimation of the sludge wasting rate. Because sudden changes in the wasting rate may create unsteady state conditions within the system, the change in the WAS rate should not exceed 10% of the current wasting rate. If the required wasting rate is below the minimum capacity of the waste sludge pumps, the hours of operation, which the pump is running, are calculated by the following equation:

$$t_{ON} = \frac{(Q_w) (24)}{Q_{p,min}} \quad (2-40)$$

where t_{ON} = Duration of waste sludge pump operations, hrs

$Q_{p,min}$ = Minimum flowrate capacity of the sludge pumps, gpm

In all cases, the operator can control the sludge wasting rate by observing MLVSS in the basins, MLVSS in the sludge line, and the organic loading to the aeration tanks.

The wet well may also play an important role in the control of the WAS rate. The operational levels of the wet well are not only restricted by the physical dimensions but the amount of RAS, WAS, and the sludge blanket level in the clarifiers. All these parameters have to be considered in the wet well operations so that there will be no inconsistency of the mass balance calculations in the system.

2.1.4.4 RAS Chlorination

One of the alternative control strategies for bulking is RAS chlorination (Neethling, 1983; Neethling *et al.*, 1984). To eliminate the filamentous organisms, the RAS is chlorinated to the levels that filamentous organisms are destroyed and enough harmless organisms are left to consume the organics. There is, however, a risk in this method. If the RAS is over chlorinated, a total loss of biomass may occur. The optimum concentration of chlorine can be found by an experienced operator after several trial and error procedures. Even though the method is risky, many treatment plants adopted RAS chlorination because it gives fairly fast results and very effective controlling strategy against the growth of filamentous organisms.

Chlorine is used widely in the disinfection of the treated water. The RAS chlorination is an attractive solution because the plant management can update the existing chlorination facilities with a minimal of capital cost. The chlorine is introduced to the RAS where there is good mixing. Generally the chlorine gas is diluted with the treated water and then by physical mixers, it is injected into the RAS line (see Figure 2-8). The amount of chlorine is a function of the amount of biomass present in the sludge (lb Cl_2 /1000 lb SS·d). Typical starting concentration is approximately 2 lb Cl_2 /1000 lb SS·d and then the dose can be raised depending on the activated sludge process response up to 9-10 lb Cl_2 /1000 lb SS·d (Jenkins *et al.*, 1982). The dose, however, is not a constant. Every plant will have different demands. The only way to determine the demand is by lab tests and by trial and error procedure. Once the necessary dose is known, the operator adjusts the chlorine gas valve and measures the concentration in the dilution water to make sure the adjustment is appropriate. If readjustment is required, the operator should observe the activated sludge process response until a steady state of chlorine demand is achieved.

The RAS chlorination is a temporary bulking control solution. It is not designed to be used continuously because the RAS chlorination has potentially high risks and high operational costs (Chambers and Tomlinson, 1982). The method should be accepted as an emergency response to critical bulking episodes. Other preventive measurements against bulking should be considered to achieve a trouble-free operation.

2.1.4.5 Nutrient Addition

One of the major causes of bulking is reported as nutrient deficiency. The only way to overcome this type of bulking is by supplying necessary amounts of nutrients to the biomass. The external addition of nutrients however is expensive. If excess nutrients are added due to poor operations, the operational costs may increase dramatically. The nitrogen, phosphorous, and iron are the most important nutrient supplements of biological cells. If any of these nutrients are absent or present in insufficient amounts, the biomass will not reproduce, thus the inadequate number of cells may not consume all the organics in the wastewater. Consequently, the excessive BOD will escape to the effluent. An adequate ratio between BOD₅, nitrogen, phosphorous, and iron should be maintained constantly to ensure the stability of the activated sludge process. The optimum ratio is given as: BOD₅|N|P|Fe is equal to 100|5|1|0.5.

Each nutrient is kept separately in chemical feed tanks. Based on the incoming BOD₅ concentration, the operator adjusts the chemical feed pumps to supply an adequate amount of nutrients to the system. The sensors in the aeration basins provide a feed back information to the operator whether the optimum ratio is obtained or not. Depending on this information the operator adjusts the speed of the pumps for the chemical feed tanks. The response, however, is not quick. The operator should take into account that there is a long lag time for the process to find a stable value.

When there is an indication the bulking episode is disappearing, the nutrient addition can be reduced by the rate of 5% per week. Close observation of sludge settleability should be performed during this period. If bulking episodes appear again, nutrient addition should be continued.

2.1.4.6 Polymer Addition

In the case of severe bulking episodes, an emergency process control strategy is taken. Perhaps treating the symptoms instead of controlling the source is not practical but it saves the management from violating the effluent standards. Temporary control actions should be applied frequently in the treatment plants because they solve the current problem quickly and effectively. Any indication of problems are generally caught by the operator during observations of the effluent quality.

If turbidity is increased in the effluent, the operator may add a chemical, namely polymers, to improve the ability of the flocs to settle (Tenney and Stumm, 1965). The addition of flocculent chemicals cause small flocs to get together and settle better. Besides the polymers, the alum is frequently used to treat the sludge bulking symptoms (Finger, 1973).

The use of ferric salts is also a common practice. Although the chemical flocculation is costly, most treatment plants have the setup to execute the method in case effluent standard violations occur. Generally, the chemical flocculation basin is not designed for the activated sludge process. Most

of the time, there is a chemical storage tank at the end of the activated sludge basin (see Figure 2-10). In case of high turbidity in the effluent, the operator may release some flocculent chemical where there is proper mixing at the weir overflow at the activated sludge channel. The chemical then has enough time in the clarifier for flocculation of the small bio-flocs.

In some conservatively designed treatment plants, there is a chemical flocculation tank after the activated sludge process and before the clarifiers (see Figure 2-10). The chemical is added to a chamber where a rapid mixing occurs. After rapid mixing, slower speed mixing chambers are commonly placed. At these lower speed mixing zones, the bio-flocs are chemically bonded and bigger flocs are produced. Consequently, better settlement of the flocs is achieved. The method is used as a temporary solution because of chemical costs. Most treatment plants have no financial sources to accommodate the cost of these chemicals.

2.1.4.7 Advanced Process Control Options

Optimum operation of the activated sludge process can only be achieved if the operator takes preventive actions against the problems. Temporary solutions are costly and they will not bring a permanent solution. The operator should identify the sources of the problems. Most of the time the long-term solutions can be applied successfully. In the case of nutrient deficiency in the wastewater, an industrial waste, which contains a high concentration of necessary nutrients, can be put in the same collection

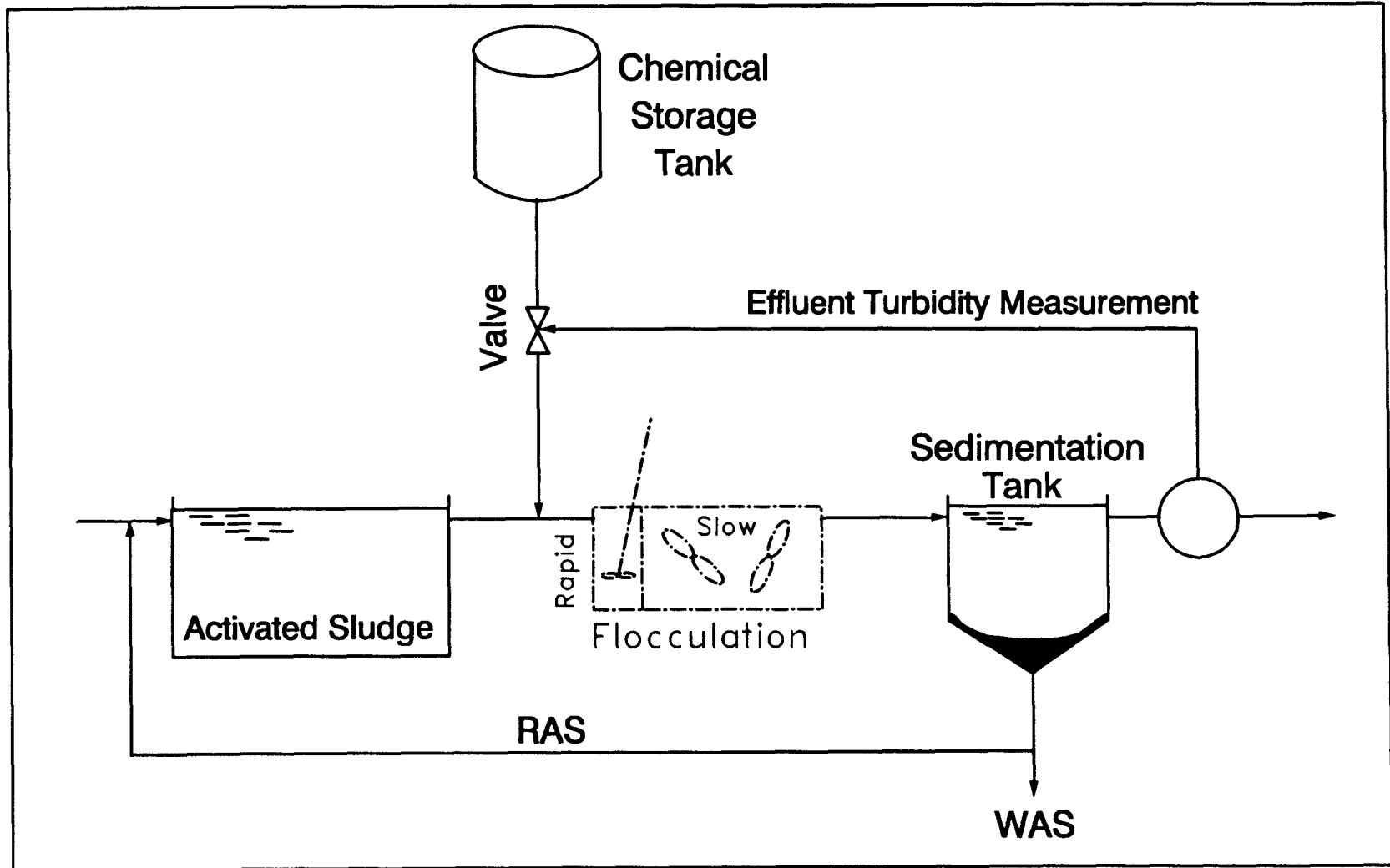


Figure 2-10: Chemical addition control scheme

system as the treatment plant. Toxic shocks can be eliminated by putting regulations on the source factories discharge permits so that the factory has to reduce its toxic discharge. The same criteria can be applied in the case of acidic/basic waste. Consequently, the pH of the influent wastewater is kept fairly constant.

It is also important that the treatment plant has the flexibility to accommodate both the chemical and hydraulic shocks. Spare equipment and flexible process modification options would be helpful to the operator to overcome the problems.

2.1.5 Mathematical Models for Activated Sludge

The ultimate process control strategy for the activated sludge process would be the use of mathematical models, if the models simulated the system with 100% accuracy. The operator would have been able to enter the measurements from the equipment readings and the model would then have predicted future events precisely.

Several researchers attempted to develop a mathematical model to simulate the activated sludge process. Some of these models simulate the organic removal mechanisms in wastewater treatment field which were included in Jorgensen and Gromiec (1985), Grady *et al.* (1986), Henze (1986), Henze *et al.* (1987a), Tang *et al.* (1987), and Van Niekerk *et al.* (1988). The oxygen transfer mechanism has an important place in the activated sludge

process. An estimation technique for the oxygen transfer capacity is investigated by Stenstrom *et al.* (1989).

One well-known model is the IAWPRC model which was developed by the task force investigating the mathematical simulation of the activated sludge process (Henze *et al.*, 1987b). In the IAWPRC model, the equations, parameters, and constants are arranged in a matrix format which are used to derive the differential equations describing the system. After its development, many applications of this model to the lab scale experiments were reported (Gujer and Henze, 1991; Sollfrank and Gujer, 1991).

The IAWPRC model was developed based on the microbial reaction kinetics. The coefficients in the kinetic equations thus greatly affect the model accuracy. Several researcher including Vaccari and Christodoulatos (1990) used an estimation technique to assign values for the kinetics coefficients in the mathematical model.

The effects of temperature and pH on the growth kinetics coefficients are identified by the works of Charley *et al.* (1980), Painter and Loveless (1983), and Antoniou *et al.* (1990). The effect of the dissolved oxygen on the microbial kinetics is also investigated by Stenstrom and Poduska (1980) and Hanaki *et al.* (1990). The microbial growth under multiple limiting-substrate is modeled by Chen and Christensen (1985), Rozich and Castens (1986), and Gee *et al.* (1990b).

Modeling nitrification in the activated sludge process has also been extensively researched. A dynamic model of the nitrification process was developed by Poduska (1973). Later, this model was applied to a number of lab experiments for the model verification by Poduska and Andrews (1975). Beck (1981) also used the model to predict the nitrification dynamics and the operational behavior of the activated sludge system. Stenstrom and Song (1991) developed a nitrification model considering the mass-transport mechanism in the activated sludge flocs and the competition between the heterotrophs and the nitrifiers.

Some other modeling techniques for the nitrification process include modeling the single-sludge nitrogen removal (Argaman and Brenner, 1986), application of a dynamic model for the activated sludge with the use of Monte Carlo simulation (Lumbers *et al.*, 1989), development of a simple model for the nitrification process (Tomei *et al.*, 1990), evaluation of the nutrient removal modeling in the activated sludge plants (Lotter and Pitman, 1990), and a review of the nitrification and the denitrification process kinetics modeling (Wentzel *et al.*, 1991). Furthermore, the nitrification process was investigated in the rotating biological contactors by Watanabe *et al.* (1990), and the prediction of the in-stream nitrification rate was accomplished by the works of Warwick and Shetty (1988).

Dynamic modeling of the activated sludge process has also been extensively studied. Andrews (1974) outlined two dynamic models for the step-feed activated sludge process and for the anaerobic digester. Dynamic modeling of the activated sludge process based on the material balance of

the stored, active, and inert fractions of the MLVSS was introduced by Busby and Andrews (1975). A complex dynamic model for the simulation of the nitrification process in the activated sludge process was developed by Poduska (1973), and Poduska and Andrews (1975).

The evaluation of the dissolved oxygen profile for the process control of the activated sludge processes was developed by Olsson and Andrews (1978). A model using the specific oxygen uptake rate (SCOUR) for real-time control of the activated sludge processes was developed by Stenstrom and Andrews (1979). A dynamic model which is used to predict the oxygen utilization was developed by Clift and Andrews (1981). The process control of the activated sludge plants based on the dynamic models was suggested by Olsson (1987).

The dynamic modeling of fixed-film growth, the settling process, the anaerobic digestion process, and the activated sludge process is comprehensively outlined in Patry and Chapman (1989). Moreover, a dynamic model of the nitrification process in the fluidized beds was developed by Stevens *et al.* (1989).

There are several additional application fields for the mathematical models in the water quality field besides the wastewater treatment plants. Some of the current studies include: the groundwater modeling (Tang and Pinder, 1977; Sykes *et al.*, 1982; Chu and Willis, 1984; Wang and Chatterji, 1984); lake and river basin modeling (Gee and MacArthur, 1981; Willey, 1982); and the water quality modeling (Roesner *et al.*,

1981; Connolly and Winfield, 1984; Barnwell and Boynton, 1985; Medina, 1985).

In addition to mathematical modelling, a number of time series analysis models were developed for the process control of the activated sludge processes (Box and Jenkins, 1976; Berthouex *et al.*, 1978; Ellis *et al.*, 1990; Novotny *et al.*, 1990).

Despite massive research of the activated sludge process, there are currently no mathematical models which simulate the process with desired accuracy. The following reasons are accounted for the failure in developing an accurate model for the activated sludge process.

The variables in the influent wastewater change constantly. These variables fluctuate so much that a steady state condition rarely exists in the activated sludge process. The fluctuations in the variables make the system behavior harder to understand. The models which are written for the activated sludge process restrict the number of variables involved in the process. A true representation of the process requires many more variables than the ones declared by the models mentioned above.

Because of the extensive number of variables involved in the microbial process kinetics, a comprehensive understanding of the process becomes difficult. Additional uncertainty is introduced by the biological systems because the microorganisms do not behave in a manner which can accurately be formulated. Although an approximation of the behavior

patterns is available for the biological systems, the results are not accurate enough to be used as a process control technique.

The mathematical models, even the most sophisticated ones, are only able to simulate the process in a limited capacity. The application of the mathematical models which have a large number of equations may require expensive computer hardware and considerable computation time. For this reason, if a mathematical model becomes too complex, the use of that model is no longer a practical alternative for real-time applications. However, if the model lacks adequate equations to describe the process accurately, the performance of the model would be very limited. The balance between a complex and a simple model represents an important decision criteria for the selection of the model.

In this dissertation, a selected mathematical model will be used to obtain numerical values for the activated sludge process parameters (see section 3.1.5.2). The primary function of the mathematical model in the expert systems is to supply numerical values for the operational variables which are used in the knowledge base. The system model describes the behavior of the process to some extent. The accuracy of the model is only enough to be used as an indicative measure.

2.2 Expert Systems

An expert system is a computer program which can access human knowledge supplied by an expert person or persons and can handle this knowledge using a "human-like" approach to make decisions. The development of expert systems can be traced to the late-1950s and the development of the computers.

Since the significant breakthroughs in computers, the ultimate goal of *Artificial Intelligence* (AI) researchers has been to duplicate with a computer human-like behavior such as speech, movement, and thinking. Research on making computers think, learn, and move like humans is rapidly developing area of AI research. The computer simulation of human phenomena first received attention in the late-1950s (Rosenblatt, 1957). Although his attempts to duplicate a human brain were limited, Rosenblatt was one of the first researchers who introduced *Neural Networks* to the AI field.

Later, in the 1960s, a new line of AI research was introduced. Exploration of the human thinking process lead researchers to define the thinking process as a manipulation of symbols. These symbols are explained as an accumulation of past learning experiences in encoded forms. Today, this type of symbol handling is called *heuristics* or in more common words, *rule-of-thumb*.

Heuristics are used to solve problems when there are no quantitative solutions. Words like "hunch" and "gut feelings" show the use of

heuristics in human decision making capability. The formulation of heuristics was first introduced in the late-1960s by Ernst and Newll (1969). These researchers developed a method called the "General Problem Solver" (GPS) to use heuristic knowledge in the computer decision making process. Although the capabilities of the GPS are limited, it has been an important step in providing guidelines for future developments in the field of heuristics.

Following the theory of heuristics, a more applicable theory, *knowledge*, was introduced into the AI field. Knowledge phenomena is explained as a series of rules which are obtained from past experiences. Knowledge is represented in the form of rules; for example, "if an apple drops, it will fall to the ground." When the rules are collected under specific headings such as everything you know about mathematics, or everything you know about art, the set of rules is called a *knowledge base*.

Knowledge theory is very attractive to computer programmers because the knowledge-based rules are easier to integrate into computer programs than "gut feelings;" however, representation of knowledge is not an easy task. Since the introduction of the knowledge concept in the early-1970s, AI researchers have continuously been working on guidelines for representation of the knowledge-based rules. Preliminary work on the guidelines has contributed to the development of the *expert systems* concept.

A computer which utilizes a knowledge base to make decisions is an expert system. Ordinary computer programs can answer questions only if the answers were previously provided. In contrast, expert system programs have the ability to produce an answer without receiving the answer prior to the question. Today, expert system programming is one of the high-level computer programming techniques.

2.2.1 Introduction to Expert Systems

This section will address the basic characteristics of an expert system. The basic objective of an expert system is to capture and store the knowledge of an "expert" person for later use by "non-expert" people. The term "expert person" here refers to a person or persons who have comprehensive knowledge in the wastewater domain. The "expert" knowledge is obtained from an expert person, combined into a knowledge base, and used to make decisions.

The decision making capability of an expert system can be described in the following steps:

- When a specific task is requested from an expert system, the expert person's knowledge is accessed through the knowledge base.
- Rules in the knowledge base are examined, manipulated, and evaluated during the decision making process.
- At the end of the examination, the rules have either one of two values: "pass" or "fail."

- Once values are assigned to all the rules, the expert system then identifies a path going through the rules which have a value of "pass." The most favorable decision lies at the end of this particular path.

Typical parts of expert systems are: an expert person, a utility to extract the knowledge from the expert person, a knowledge manipulation box (*i.e.*, inference engine), and a human-machine interface (see Figure 2-11). The knowledge of the expert person is extracted by the questions asked during interviews with the expert person. The extracted knowledge is then processed by the knowledge manipulation box, more commonly known as the inference engine. The decision making capability of the expert system lies inside the inference engine. The human-machine interface allows the non-experts to access the knowledge stored in the expert system with a user-friendly interface.

Knowledge extraction is called *knowledge acquisition* in the AI field. Because a significant amount of work is involved in extracting heuristic knowledge, the knowledge acquisition is considered to be the most difficult aspect in the development of expert systems. One of the setbacks is that experts cannot explain their actions accurately because they just *know them by intuition*.

A primary objective of the expert system developers is to capture an adequate amount of knowledge from the expert so that the expert systems will have a decision making capability equal to that expert. It is essential to formulate the knowledge into a rule form within the knowledge base.

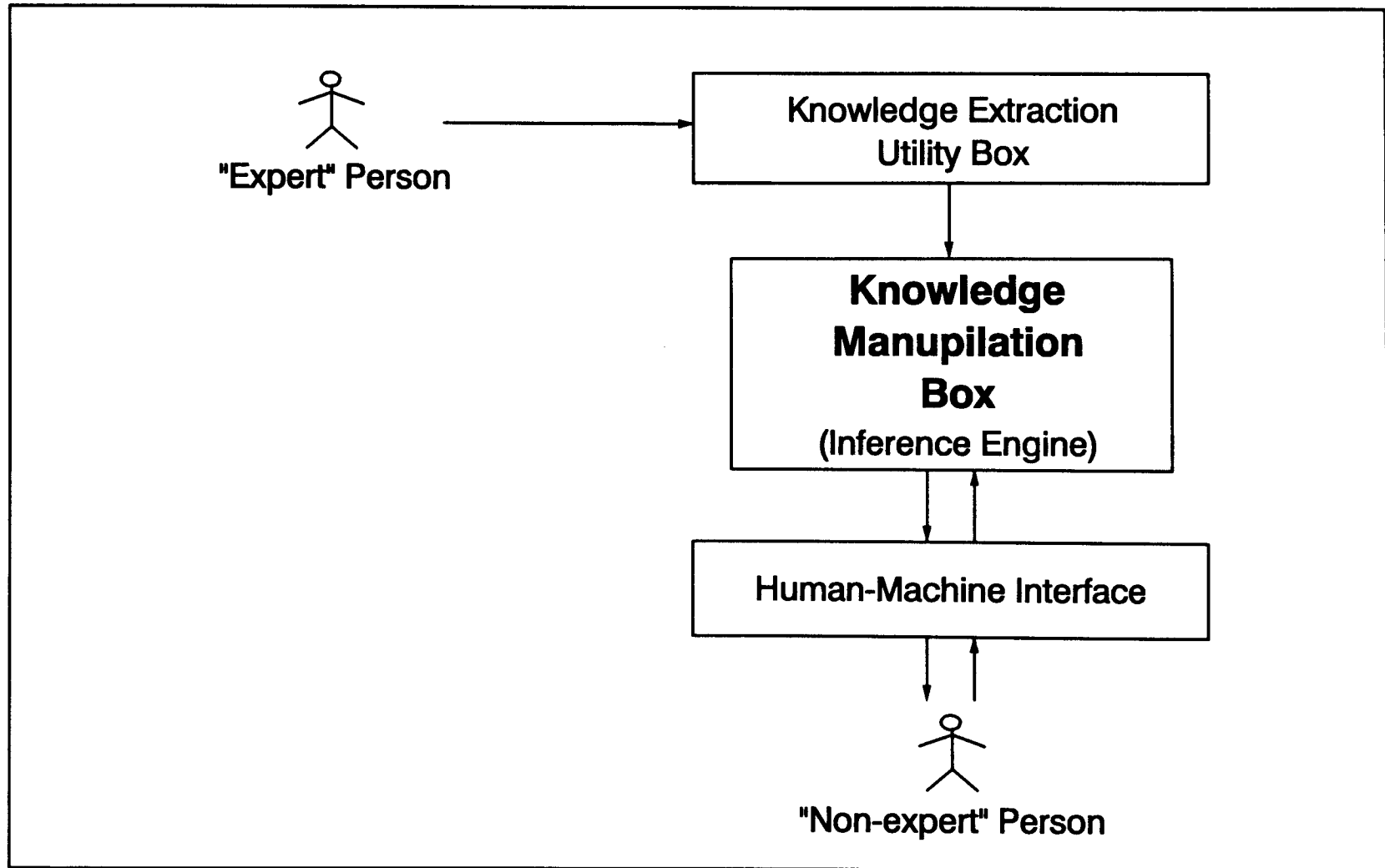


Figure 2-11: Basic components of an expert system

Once all the rules are compiled into a knowledge base, the expert system is capable of manipulating this set of rules during the decision making procedure. The rules are entered into the knowledge base in a form by which they can be efficiently organized and manipulated by the inference engine. The most common way to represent knowledge in a rule form is using *IF* (...) and *THEN* (...) statements.

The *IF* (...) portion of a rule contains "conditions." These conditions in a rule are often called the *premise*. The second section of a rule, the *THEN*(...) clause, suggests a *conclusion*. Every rule must contain both of these components. For instance, the control of dissolved oxygen (DO) in aeration basins can be formulated as:

RULE 007

IF DO in aeration basin is less than 1.0 mg/L,
THEN Turn air flow control valve completely to "ON" position.

The knowledge base is comprised of rules similar to the one shown above. A lab prototype knowledge-based expert system with a reasonable coverage has approximately 400 rules. More extensive coverage, however, may require as many as 1,400 rules (Wigan, 1986). The structure of a knowledge base can be illustrated as follows:

RULE 63

IF Condition ABC
THEN Take ACTION 123

RULE 64

IF Condition XYZ

THEN Take ACTION 999

. .

. .

RULE 100

IF Condition RED

THEN Take ACTION 911

Every rule in the knowledge base is evaluated by the expert system. The evaluation process is commonly called *firing a rule*. Once the rule is fired, the *IF* (...) segment (premise) is analyzed. If the premise is satisfied, the rule is marked "passed." Once a rule has passed, the *THEN* (...) segment (conclusion) is executed. The conclusion of a rule can either be a specific action or an assigning value to a variable. If a premise is not satisfied, the rule is labeled "failed." A failed rule's conclusion segment is not executed and the rule is excluded by the decision making process.

It is useful to include "How" and "Why" explanations for every rule. During the rule evaluation, the user may be informed about the progress of the decision making procedure, in order not to feel insecure about accepting the expert system's conclusion.

The structural characteristic of expert systems is a current research area and many researchers are working on the expert system programming strategies. Today, there are many different expert system shells available. In the following section, computational properties, structure, and methods

of development for the expert systems are discussed along with a detailed analysis of knowledge acquisition techniques.

2.2.2 Structure of Expert Systems

This section will explain in detail the structure of expert systems. The steps necessary to develop expert systems are summarized as follows and were used in this dissertation:

- Define software and hardware environment.
- Write expert systems program code or obtain an expert system shell.
- Develop user-friendly interface.
- Collect literature rules about the domain (create a draft knowledge base).
- Interview expert persons to enhance the rules in the draft knowledge base (intermediate knowledge base).
- Test the expert systems with real-time applications and the input from expert persons (final knowledge base).
- Put expert systems into real world practice.

All the steps need to be carefully taken before expert systems can work accurately. Each step is closely investigated in the following sections.

2.2.2.1 Expert System Languages

Expert systems are mainly developed by *knowledge engineers*. The most common computer languages used for expert systems programming are **LISP** and **Prolog**. These two languages have proven their desirability over other high-level computer languages because of their ability to process symbols. A symbol or "word" is a group of characters which can represent notions or things.

Common algorithmic languages are limited to recognizing a "word" as either a series of characters or numerical values. Although it is possible to use algorithmic languages such as PASCAL, FORTRAN, and C, to create an expert system, the programmer has the disadvantage of not having symbolic computation. For this reason, it is more common to program expert systems with a symbolic language.

The LISP was developed by McCarthy (1958). The goal of the language was to solve mathematical logic problems. The language has evolved through many modifications to become useful for expert systems' applications (Abelson *et al.*, 1985). The major difference between the LISP and other algorithmic languages is that the LISP was developed to handle symbolic terms.

The second most common symbolic language in the AI field is Prolog. Prolog is considerably newer. It was developed in France in the mid-1970s (Roussel, 1975). Although Prolog is similar to LISP in its use of symbolic logic, the two languages have significant differences. LISP is

known to be easy to read, write and modify. In contrast, Prolog is much more difficult to master. Careful planing is required before any Prolog code can be written (Clocksin and Mellish, 1981). Finally, Prolog has less support graphics and data exchange operations. Prolog is still in the developmental stage and is not standardized.

The recent development of **POPLOG**, which is a programming environment, shows potential for a future tool in the improvement of expert systems (Edmunds, 1988). Future development tools will have powerful computational capabilities and even inexperienced persons will be able to use these tools without extensive computer background.

2.2.2.2 Hardware Requirements

Another important issue in developing expert systems is the suitable choice of hardware. Surprisingly, expert systems do not require expensive super-computers, and there are some expert systems that can run on machines which have as small as 64 KB memory. Personal computers (PC) are capable of running powerful expert systems which can handle the needs of today's businesses (Edmunds, 1988). Within the PCs, IBM's XT, AT, PS/2, and Apple Computer's Macintosh are all capable of running considerably complex expert systems. A 80386-based PC with at least 8 MB RAM is an inexpensive solution to many applications of expert systems. Both the AI software and the necessary peripherals for the PCs are widely available today.

Data exchange is another advantage in using PCs because many potential expert systems' users already have PC-based data bases and worksheet programs. The data within these programs can be interchangeably used along with the expert system's program.

Another advantage of PC-based expert systems is their ease of use. Although PCs limit the use of large knowledge bases and highly complex expert system programs, for many applications in the industry, a few hundred rules are sufficient for an effective expert system.

One of the most well-known special-purpose computers is the *LISP Machine* (Bowerman and Glover, 1988). The operating system is optimized for LISP. Because the machine is designed just for LISP, the program code written in LISP runs very fast. Furthermore, the output of the program is displayed by a high-resolution graphic screen to help the user more efficiently view several windows at a time. The LISP machines, however, are large and expensive machines.

Other alternatives for expert system tools include AI workstations, mainframes and super-computers. All these alternatives have the disadvantage of being large-sized computers which require advanced computer skill and are substantially more costly (Edmunds, 1988).

2.2.2.3 Expert System Shells

The core of an expert system is the inference engine which deduces logic through the rules entered into its knowledge base. A computer program which contains an inference engine and is capable of reasoning through a knowledge base but does not itself contain a knowledge base is called a "shell." The shell and the knowledge base are separate modules. The shell is a computer program which is written to manipulate the rules in the knowledge base. The shell is thus capable of running different knowledge bases one at a time.

Expert system shell functions are the same during the analysis of any knowledge base. In addition to the inference function, the shell also creates a development environment for writing the knowledge base rules and for testing the knowledge base. The type of expert system program is determined by the content of the knowledge base and not the type of shell.

Early expert system shells were large programs and could only run on main-frame computers. One of the frontier shells mentioned here is EMYCIN (Van Melle, 1979). EMYCIN is the expert system shell for MYCIN (Shortliffe, 1976). MYCIN has demonstrated a successful application of expert system technology in the medical field. The MYCIN expert system is discussed further in section 2.2.5.2.

The EMYCIN shell consisted of a user-interface and an inference engine. Normally, an expert system shell is chosen and then the knowledge base

is developed under that chosen expert system shell environment. However, in this case, MYCIN was first developed and then EMYCIN was separated from the knowledge base as the stand-alone shell. After the shell was extracted, EMYCIN was used for different knowledge bases such as SACON for the finite element of the structural analysis (Bennett and Engelmores, 1979) and PUFF for diagnosis of medical problems based on the interpretation of preliminary function data (Atkins *et al.*, 1984).

Further developments of the EMYCIN expert system shell introduced a new shell language called ROSIE (Fain *et al.*, 1981). Although EMYCIN was a base for all the above mentioned expert systems, the shell was difficult to use. Under the development environment of the shell, considerable experience with LISP was needed to create a knowledge base. Without the LISP experience, writing a knowledge base with EMYCIN takes considerable time and effort.

After the Prolog language was introduced, a number of expert system shells were developed using this new language. Prolog is closely correlated to expert system programming because Prolog's structural functions highly resemble expert system functions. Thus, Prolog became a widely used language in which to write expert system shells.

Although LISP and Prolog are the predominate languages used for writing expert system shells, other languages such as PASCAL, BASIC, and C are also used. Table 2-16 lists several expert system shells available for small microcomputers. The list contains some of the shells written in LISP,

Table 2-16: Some of the expert system shells available for personal computers (partly compiled from Wigan, 1986)

Programming language	Name of shell	Vendor	Operating system
LISP	Personal Consultant	Texas Instruments Inc.	DOS
	Personal Consultant Plus	Texas Instruments Inc.	DOS
	S-I	TeKnowledge	UNIX
	Series-PC	SRI Inc.	DOS/UNIX
	XSYS	California Intelligence	DOS
	DAISY	Lithp Systems	DOS
	IN-ATE	Automated Reasoning	DOS/UNIX/Mac
Prolog	BUILD	Sydney University	DOS
	ES/P Advisor	Expert Systems Int'l	DOS/VMS
	GEM	CSIRO	PROLOG
	MI	TeKnowledge	DOS
	PROVE	CSIRO	DOS/UNIX
	APES	Logic Based Systems	DOS/UNIX
Other: PASCAL, BASIC, and C	AION	AION Corp.	DOS
	AL/X	Intelligent Terminals	UCSD
	Expert Edge	Helix Expert Systems	DOS
	Expert System	PPE Inc.	BASIC
	EXSYS	EXSYS Inc.	DOS
	INSIGHT 1	Level 5 Research	DOS
	INSIGHT 2	Level 5 Research	DOS
	Level5	Level 5 Research	DOS/Mac
	Level5 Object	Level 5 Research	MSWindows
	KES	Software A&E	DOS/UNIX
	KnowledgePro	Knowledge Garden Inc.	MSWindows
	MicroEXPERT	ISI	DOS/VMS
	NExpert Object	Neuron Data Inc.	MSWindows
	REVEAL	McDonnell Douglas	DOS
	SAGE-PC	SPL-Insight	DOS/UNIX
	SAVOIR	ISI	DOS/UCSD
	Symbologic Adept	Symbologic Corp.	MSWindows
	VP-Expert	Paperback Software	DOS
	Wizdom	Software Intelligence	DOS
	XL	Intelligent Systems	UNIX

Prolog, PASCAL, BASIC, and C. With the growth of expert systems technology, the list of shells is rapidly expanding. Today, the primary task is not to find a shell but to choose a suitable shell for the given application.

Choosing the most suitable shell for any given application depends on the type of application being performed and the functions required from the expert system shell. Figure 2-12 shows the evaluation sheet used when choosing the most suitable expert system shell for our application. The evaluation sheet covers three major topics: software requirements, hardware requirements, and vendor support.

Figure 2-12 gives a complete list of software and hardware requirements. Some of the software requirements include ease of rule writing, the ability to direct rule firing sequence, the ability to use an extended number of rules (over 300 rules), the ease of arranging rules, the ability to use external programs, and the ability to generate an individual run-time copy of the knowledge base. Overall, the software should be easy to use and should have advanced options to handle complicated expert system applications.

In addition to software and hardware requirements, the support provided from the vendor plays a key factor in deciding on which shell to purchase. User support groups, supporting documentation, and training materials help users overcome problems during the development of the knowledge-based systems.

<u>Software Requirements</u>	<u>Hardware Requirements</u>
<input type="checkbox"/> Simple command structure (no-memorize commands) <input type="checkbox"/> Review more than one rule at a time <input type="checkbox"/> Comprehensive error messages <input type="checkbox"/> Ease of rule organization <input type="checkbox"/> Ability to change rule number <input type="checkbox"/> Ability to trace specific frames and rules <input type="checkbox"/> Fast switching between editing and running <input type="checkbox"/> Ability to handle over 300 rules <input type="checkbox"/> Easy variable assignments <input type="checkbox"/> Easy access to multi-format DOS files and programs <input type="checkbox"/> Ability to generate a run-time copy of knowledge base <input type="checkbox"/> Ability to manipulate graphical images <input type="checkbox"/> No restricted number of variables <input type="checkbox"/> Ability to stop and resume during tracing <input type="checkbox"/> Graphical tracing capabilities <input type="checkbox"/> Ease of changing variable values during tracing (what-if) <input type="checkbox"/> Ability to alter rule content during tracing process <input type="checkbox"/> Ability to perform complex mathematical functions <input type="checkbox"/> Price, inexpensive upgrade option	<input type="checkbox"/> Ability to operate on a PC <input type="checkbox"/> Not more than 2MB memory requirements <input type="checkbox"/> Not more than 5MB hard-disk requirements <input type="checkbox"/> Ability to run with limited memory <input type="checkbox"/> Inexpensive add-ons <input type="checkbox"/> Ability to run fast enough with slow PCs <input type="checkbox"/> Inexpensive hardware requirements <input type="checkbox"/> Ability to use extended/expanded memory
	<u>Vendor Support</u>
	<input type="checkbox"/> User support toll-free hot-line <input type="checkbox"/> Common Q&A sheets <input type="checkbox"/> Training courses <input type="checkbox"/> User support groups <input type="checkbox"/> Fast turnaround time <input type="checkbox"/> Support documentation provided <input type="checkbox"/> Public modem access to its bulletin board <input type="checkbox"/> Ability to download sample KB via modem <input type="checkbox"/> Published periodicals which give examples

Figure 2-12: Requirements relevant to choosing a shell

2.2.2.4 Manipulation of Logic

Knowledge in expert systems can be represented in many different ways (Brachman and Levesque, 1985; Boger and Guterman, 1990; Giuse, 1990). The most common styles are semantic networks, modifications of object-attribute-value, frames, logic trees and rules (Harmon and King, 1985; Edmunds, 1988). Combinations of several different techniques are also used and are commonly called hybrid systems.

It is argued that *semantic networks* are the closest model to human memory functions (Quillian, 1968). Semantic networks combine groups of information and develop a relationship between the concepts. Semantic networks are commonly illustrated in graphical formats which are called diagrams. A semantic network diagram has "objects," "nodes," and "links" (Alty and Coombs, 1984; Edmunds, 1988). The **objects** represent a thing such as a place, physical material, person, or idea. The objects are represented by **nodes** in the diagram. Nodes are connected to each other by **links**.

Links are used to show the relationship between the nodes (*i.e.*, objects). Phrases like "*is-a*" and "*caused-by*" are often used to describe the association between two nodes. A typical component of a semantic network is illustrated in Figure 2-13(a) and an example of a semantic network is given in Figure 2-13(b).

Formulation of knowledge with semantic networks is simple. Each node is related to each other with a phrase. By analyzing the network structure

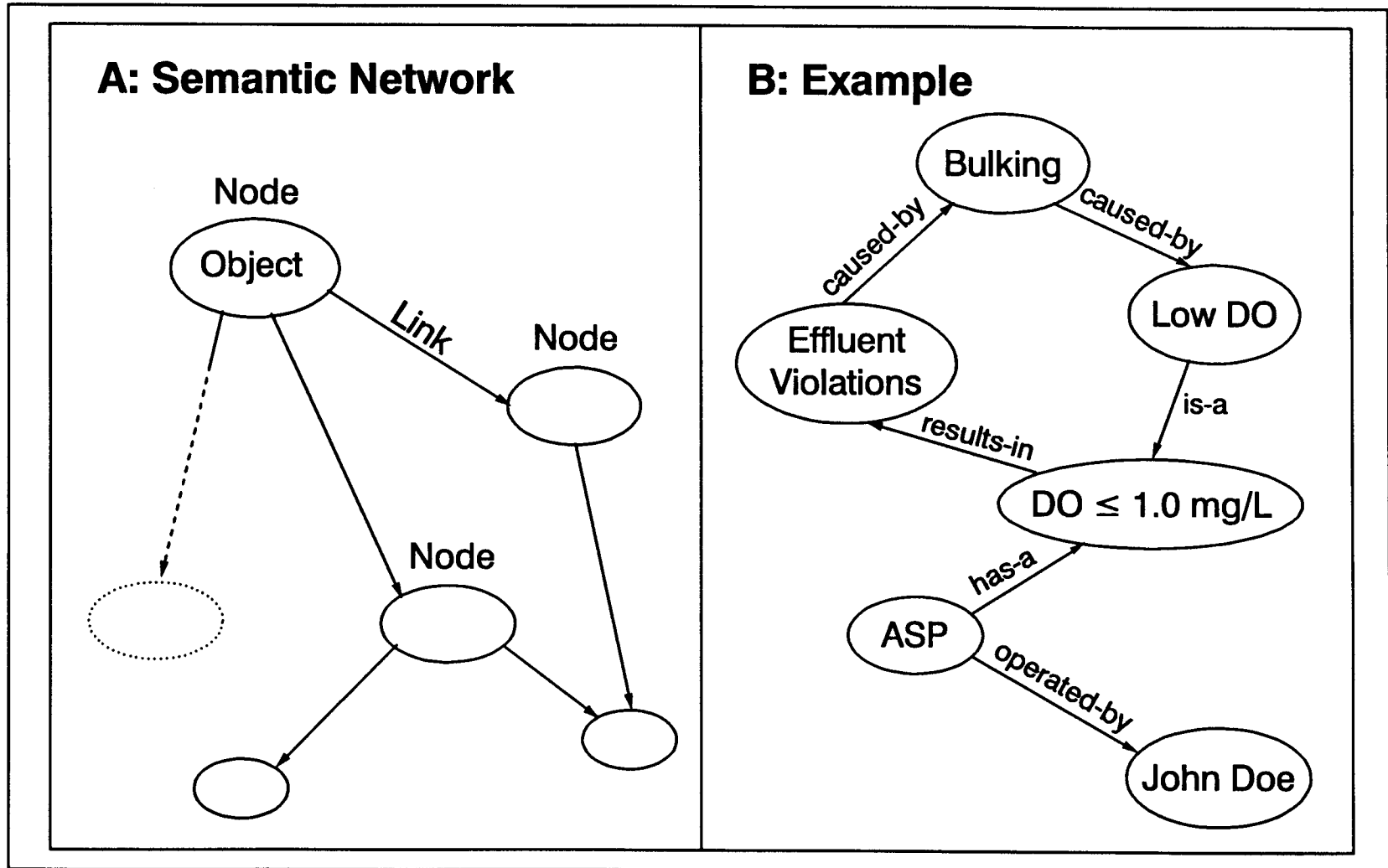


Figure 2-13: (a) Knowledge representation by semantic networks; (b) Example of semantic networks

closely, the user can easily identify existing relationships between the objects. For instance, in Figure 2-13(b), the cause of effluent violations is identified as bulking. Moreover, it can be seen that the bulking is caused by low DO or when the DO is less than 1.0 mg/L. From this diagram, if the activated sludge process (ASP) has a low DO, then the results would probably be a violation of effluent standards due to bulking.

Similar relations can produce many possible paths within the semantic networks. The computer analyzes every possible path on the way and designates a path that will lead to the optimum solution. Because semantic networks are easy to develop and use, expert system developers have adopted the method widely.

One modification of a semantic network method is called *object-attribute-value*. This method is similar to the semantic networks. **Object** is used for an explanatory name of a thing or concept. The object is recognized as a symbol during the knowledge base manipulation. Every object is coupled with an **attribute** which is used to describe the object. An attribute can be such characteristics as color, shape, physical dimensions, or appearance. An attribute is used to describe the properties of an object. Every attribute has a **value**. For example, attribute is size and value is large, or attribute is price and value is \$10. Combining all three provides a helpful description of an object. For instance, activated sludge foam (object) color (attribute) is dark brown (value).

A typical knowledge base will have many object-attribute-values in its structure. Each of these are connected by links that describe the relationship. A simpler method also exists of combining only the attribute and value pairs. However, this technique is not used in expert systems because of its limitations for assigning object names.

A more complex knowledge presentation is performed by the *frame-based* systems. The concept of frame-based systems were introduced by Minsky (1963). Despite their complexity, the frame-based systems are powerful tools for knowledge manipulation. One of the early examples of a frame-based system was developed by Brachman and documented by Brachman and Schmolze (1983).

A typical frame-based system is made of slots. Each slot contains information about an object. The first slot in the frame is normally a descriptive object name. Additional slots, which have related properties to the first slot, are attached to each other. All these slots make up a frame. A frame, therefore, contains all related properties of an object, such as values, descriptions, introductions to perform a specific action, and attributes.

Frames are connected to each other in a manner similar to the semantic network structure. Once the related frames are identified, the information is exchanged within the frames by creating a path along the way. Repeating this step creates a tree of frames where the slots of individual

frames are connected to each other. Slots give frame-based systems a powerful opportunity to represent knowledge accurately.

Proper organization of frame-based systems also provides additional help to the expert system developer. Objects are kept under specific group names. As an example the first frame could be properties of microorganisms with the following sub-category being aerobic organisms and anaerobic organisms. For an illustration of the frame-based knowledge presentation see Figure 2-14. The activated sludge problem frame is matched with the microorganisms' frame. In this manner the problems and the possible causes can be identified by analyzing the frames.

Use of logic trees is another way of representing knowledge. Logic trees are constructed by mathematical logic principles. The logical representation is quite simple. One well known example of a logic statement is: "All persons are mortal. Socrates is a person. Therefore Socrates is mortal" (Kalish *et al.*, 1980). Most relationships are deduced in a similar manner.

Another important property of a logic statement is a value of true or false. In computer programs, this logic is symbolized by "1" for true and "0" for false. Once all the statements are analyzed, the computer creates a series of "1"s and "0"s. The "1"s also show a possible path for an optimum solution. A comparison of multiple paths in a logic tree provides an optimum solution for a problem.

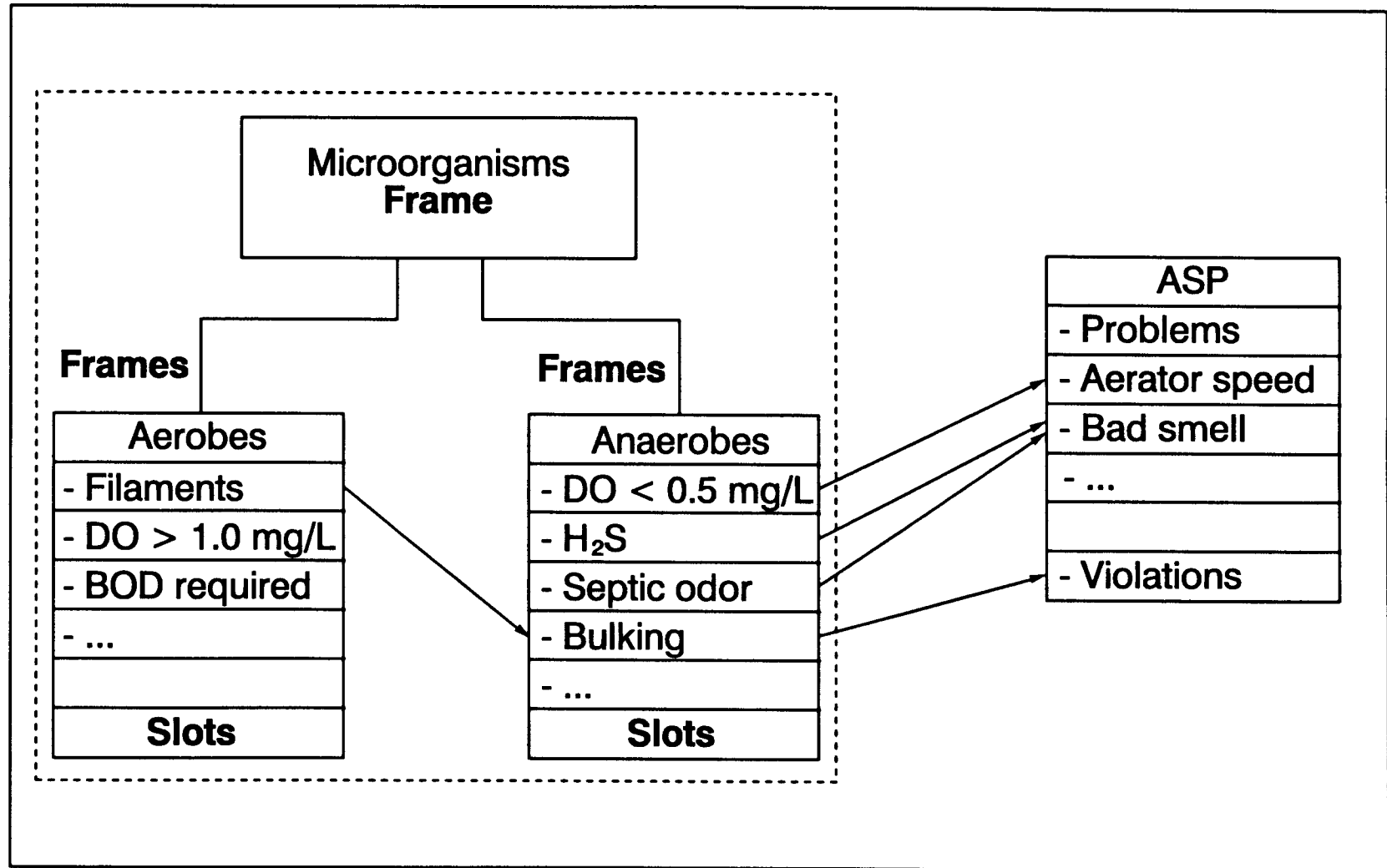


Figure 2-14: Knowledge representation by frame-based systems

Connections between statements are achieved by phrases like **AND**, **OR**, and **NOT**. These coordinating conjunctions have an important role in describing the relationship between the logic statements. For instance, the following logical statements demonstrate the use of logic.

Facts:

- ★ Activated sludge is a wastewater treatment process.
- ★ Biological systems contain microorganisms.
- ★ Some microorganisms only survive in aerobic conditions.
- ★ Activated sludge has many microorganisms.
- ★ The microorganisms in the activated sludge need oxygen to survive.

Logic:

- ★ Activated sludge system is an aerobic biological wastewater treatment process.

The most common knowledge presentation in expert systems is *knowledge-based* systems. Knowledge is presented in a rule form. Thus, the knowledge-based systems are commonly referred to as rule-based systems. Rules, as explained earlier in the text, are made of two major components: *IF* (...) and *THEN* (...). For example, a rule for controlling aerators in an activated sludge plant can be written as:

IF There is violent turbulence in the aeration basins,
THEN Reduce air flowrate.

The use of rules can be ambiguous. If an inadequate number of premises are included in the *IF* (...) block, the final decision may be misleading. For example, the rule above gives the wrong recommendation if there is low DO in the basins. A possible problem can be eliminated by introducing more conditions in the premise section. One way to do this is the use of "AND," or "OR" words. The example shown above can be expanded as:

IF There is violent turbulence in the aeration basins,
AND The DO is high in the basins,
THEN Reduce air flowrate.

Other necessary conditions may be added to the premise block to make the rule more specific. However, it should be noted that more conditions do not assure higher reliability of the solution. Sometimes it is wise to leave the rules as general as possible because too specific conditions may never exist. Thus, the rule may never be used. The optimum number of conditions in a rule is decided by the expert system developer. The quality of the rules is associated mainly with the developer's expertise about the knowledge representation.

2.2.2.5 Knowledge Acquisition

In the development of expert systems, the knowledge acquisition is the most important stage. And because knowledge is acquired mainly from

the expert persons' heuristic experience, knowledge acquisition is also the most difficult task. The basic goal of an acquisition process is to capture the experience that the expert person has accumulated. The basic steps in the knowledge acquisition are as follows:

- Knowledge engineer performs literature search to learn more about the subject.
- Knowledge engineer defines rules/laws as a result of literature search.
- Knowledge engineer makes a list of crucial questions to ask expert person.
- Knowledge engineer interviews expert person who has demonstrated high skill performance in the domain.
- Knowledge engineer combines the rules from literature and from the interview to create a knowledge base for the domain.

The acquisition process is executed through interviews between the knowledge engineer and the expert persons. Knowledge engineers face difficulties when they deal with heuristic knowledge. Most of the expert persons have trouble explaining their ideas. Frequently, they respond "I just know it," or "I have a feeling about it." For the expert system developer, the heuristic knowledge is difficult to formulate because the computer does not have the same intuitions as the expert persons have. Figure 2-15 illustrates the efficiency of knowledge acquisition.

In Figure 2-15, part of the knowledge is lost during the acquisition process steps such as interviews, knowledge formulation, and knowledge

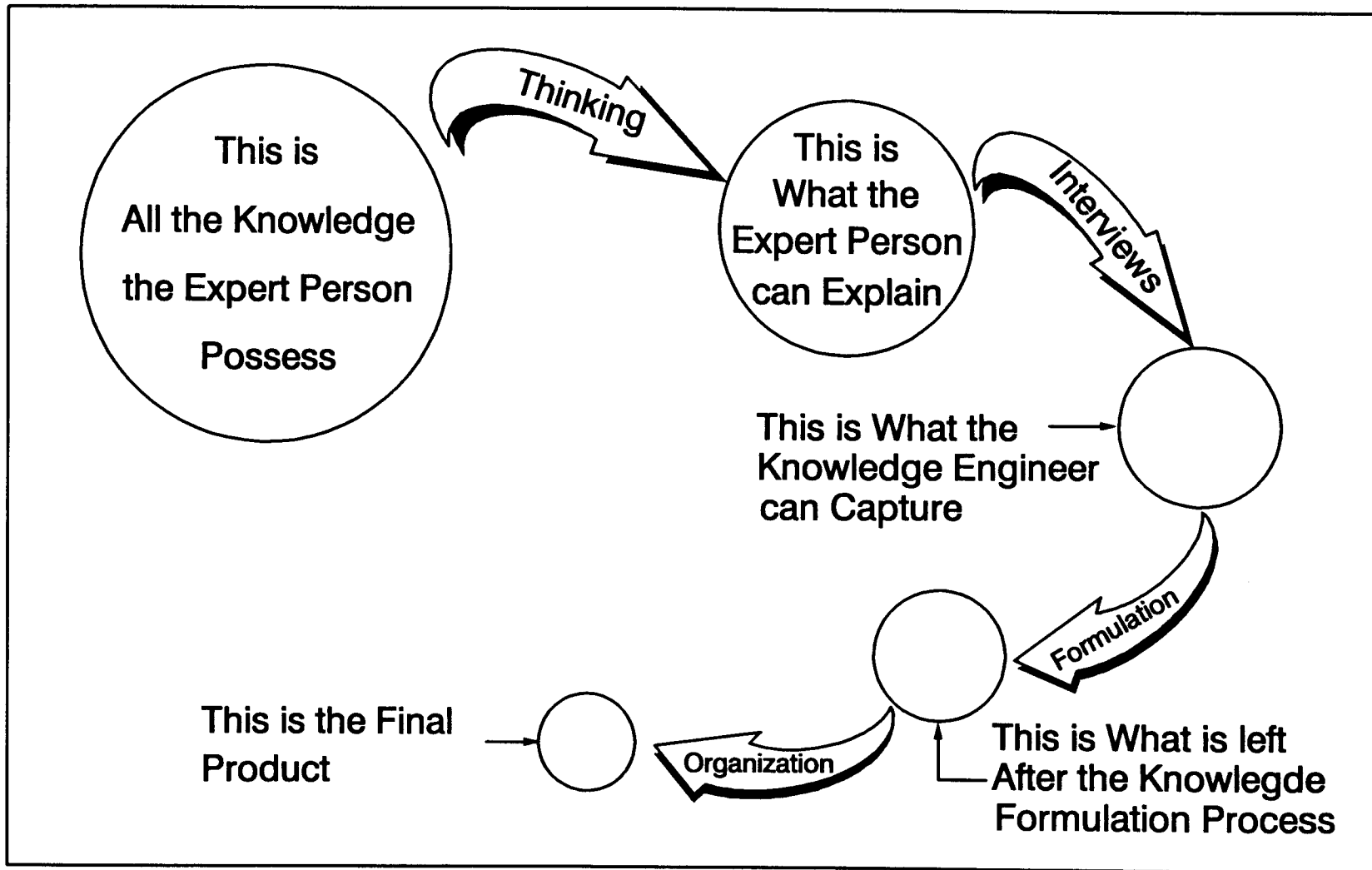


Figure 2-15: The loss of knowledge during knowledge acquisition

organization. Furthermore, at the beginning of the knowledge acquisition process, transference of the expert person's primary knowledge is limited by the expert person's ability to completely explain what he/she knows. During the interview sessions, additional knowledge is lost due to inadequate communication between the expert person and the knowledge engineer. Because asking the right questions is equally important as giving the right answers, the knowledge engineer should be familiar with the expert's field. More knowledge is lost during the knowledge formulation process because of incomplete knowledge formulation tools. Finally, additional information is lost during the knowledge organization process due to the absence of accurate knowledge organization techniques.

After the knowledge acquisition process, the knowledge base is compiled. The individual knowledge pieces from the interviews and the rules from literature are checked for consistency. Once the acquired knowledge is confirmed, the draft knowledge base is ready to be constructed.

The idea of using another expert system to capture the "strategic" information during the knowledge acquisition process is in a premature state. Further investigation of the possibilities may reveal an effective tool to extract knowledge from the expert persons. If this tool is successfully developed, inexperienced interviewers can take advantage of this expert interviewer tool. Development of such a tool is, however, not in the scope of this dissertation. The potential of having such a powerful tool during the knowledge acquisition process should not be overlooked. The

future development of such a tool would be very helpful to expert system's developers.

2.2.2.6 Knowledge Base Search Techniques

Once the knowledge is collected by the knowledge acquisition techniques, the second challenging task is to put the knowledge in a form that the computer can manipulate. Organization of knowledge within the knowledge base affects the performance of the expert system.

In order to detect which rules are related to the subject, the computer needs a search algorithm to determine which rules are to be processed and how they are to be processed. There are two methods for performing a search within a knowledge base. When the conclusion block of a rule is analyzed first it is called backward chaining. Alternatively when the premise block is examined first it is called forward chaining. A combination of both (*i.e.*, mixed chaining) is also commonly used. Each search algorithm is discussed in detail to reveal advantages and disadvantages of the individual methods.

Backward chaining, which is also called the backward reasoning or depth-first search technique, is a method where the "goals" are searched first (Pedersen, 1989). In other words, the computer reasons through the rules by directing the search towards a specified goal. The "goal" is a specific parameter which is being sought by the expert systems.

The following example illustrates the use of backward chaining. In the example, the knowledge base consists of 5 rules which use the dissolved oxygen concentration and the appearance of the water surface in the basins to diagnose and correct aeration problems. The goal is to search for a "Blower Rate" value when the DO is less than 1.0 mg/L and the aeration basin surface appears wavy. The search algorithm is explained step by step how the inference engine analyzes and evaluates the rules.

STEP I. Backward Chaining

Given: DO is less than 1.0 mg/L and
aeration basin surface is wavy

Find GOAL: Blower Rate

Inference Engine

Current GOAL: Find Blower Rate

Action 1: Which rules conclude the current goal?

Knowledge Base

Rule 1

IF DO \leq 1.0 mg/L
THEN DO Level = Low

Rule 2

IF Surface = Calm
THEN Turbulence = Normal

Rule 3

IF Surface = Wavy
THEN Turbulence = Violent

Rule 4

IF Turbulence = Violent
AND DO Level = High
THEN Blower Rate = Mid

Rule 5

IF Turbulence = Violent
AND DO Level = Low
THEN Blower Rate = High
Advice "Repair diffusers
immediately"

In Step I, the inference engine reads the given input, in this example, DO is less than 1.0 mg/L and aeration basin surface is wavy. The user asks the expert system what the optimum blower rate is. The inference engine recognizes this request as the current goal and sets the current goal value to "Blower Rate." In the action block of the inference engine, first the rules which conclude the current goal are searched. The inference engine checks the *THEN* (...) block of the RULE 1 and finds that this rule does not conclude the current goal. The inference engine then puts RULE 1 aside. This search is performed until the right rule is found which is RULE 4 in this example.

Once the inference engine finds RULE 4, it checks the premise of this rule. RULE 4 has two premises. The first one is a condition of "Turbulence" and the second one is a state of "DO Level." The inference engine evaluates the first premise. The value of "Turbulence" is not given by the user. The inference engine has to allocate a value for the condition in order to evaluate the premise. Now the inference engine assigns the current goal value as "Find Turbulence." The previous goal has been set aside until the new goal is allocated.

STEP II. Backward Chaining

Inference Engine

Current GOAL: Find Turbulence

Previous GOAL: Find Blower Rate

Action 1: Which rules conclude the current goal?

Action 2: Find the value of Turbulence

Action 3: Which rules conclude "Turbulence"?

Knowledge Base

Rule 1
IF DO \leq 1.0 mg/L
THEN DO Level= Low

Rule 2
IF Surface= Calm
THEN Turbulence= Normal

Rule 3
IF Surface= Wavy
THEN Turbulence= Violent

Rule 4
IF Turbulence= Violent
AND DO Level= High
THEN Blower Rate= Mid

Rule 5
IF Turbulence= Violent
AND DO Level= Low
THEN Blower Rate= High
Advice "Repair diffusers
immediately"

The second step is to assign a value for "Turbulence." The inference engine searches the rules and locates the concluding rule. RULE 2 is evaluated by examining the premise part. The value for "Surface" is given by the user as "Wavy." The inference engine checks whether the premise is true or not. Because the condition of "Surface" is assigned to "Calm," the premise is false and the rule fails. Thus, the conclusion of RULE 2 is not valid.

The inference engine continues the search for "Turbulence," and finds it in RULE 3. Now RULE 3 is evaluated. The RULE 3 is passed because the premise of the rule "Surface= Wavy" is correct. The conclusion of the rule is fired and "Turbulence" now has a value of "Violent." Since the inference engine has a value for "Turbulence," RULE 4 can be evaluated. The first premise of RULE 4 is true and the engine continues with the second premise in RULE 4. Again, the value of "DO Level" is not

known. The inference engine assigns the current goal to "Find DO Level."

By a similar procedure, the computer first finds RULE 1 and evaluates the premise as true. Consequently, the conclusion part of RULE 1 is executed. The value of "DO Level" is assigned as "Low." The inference engine goes back to RULE 4 to see if the second premise is satisfied but it fails. Because two premises in RULE 4 are connected by "AND" with one premise true and the other premise false, the total value becomes false and the rule fails. The next step is to locate more rules which will satisfy the original goal.

STEP III. Backward Chaining

Inference Engine

GOAL: Find Blower Rate

Action 1: Which rules conclude the current goal?

Action 2: Find the value of Turbulence

Action 3: Which rules conclude "Turbulence"?

Action 4: Evaluate RULE 2

Action 5: Check premise in RULE 4

Action 6: Search for another rule which concludes current goal

Action 7: Evaluate RULE 4

Action 8: Set current goal to "DO Level"

Action 9: Evaluate RULE 1

Action 10: Check RULE 4 for second premise

Action 11: Reset current goal to original goal

Action 12: Search for more concluding rules

Knowledge Base

Rule 1
IF DO \leq 1.0 mg/L
THEN DO Level= Low

Rule 2
IF Surface= Calm
THEN Turbulence= Normal

Rule 3
IF Surface= Wavy
THEN Turbulence= Violent

Rule 4
IF Turbulence= Violent
AND DO Level= High
THEN Blower Rate= Mid

Rule 5
IF Turbulence= Violent
AND DO Level= Low
THEN Blower Rate= High
Advice "Repair diffusers
immediately"

Finally, the inference engine locates RULE 5 which is concluding the condition of "Blower Rate," and fires this rule. Two premises in RULE 5 are true, thus, inference engine executes the conclusion of RULE 5. This multiple conclusion says that the blower rate should be kept high because the DO level is low. It advises the user to repair the diffusers because there is uneven air distribution which is causing violent turbulence in the aeration basin.

It is important to note that every variable in the knowledge base such as "Turbulence," and "DO Level" has attribute status which means the variable is either KNOWN, UNKNOWN or NOTSOUGHT. At the beginning of the expert system consultation, all the variables' attributes are set to NOTSOUGHT because the inference engine has not searched for these variables yet. When the user assigns values for the variables, the

attributes for those specific variables are set to KNOWN. However, if the inference engine is unable to find any kind of value for the variable, then the attribute becomes UNKNOWN.

The example above illustrates the backward chaining. The method searches the original goal by updating the current goal until the original goal is satisfied. This method is the most common procedure in expert systems. Many expert system shells have adopted the backward chaining method because it is easy to formulate and also it is an effective tool for analyzing rules.

Forward chaining is significantly different from the backward chaining. Basic algorithm in forward chaining is "start with the known facts and try to infer the conclusions implied by those facts" (Pedersen, 1989). Forward chaining is also known as breath-first search technique. Once all the facts are fed to the inference engine, rules which are related to given facts are analyzed. To compare the strategy of backward and forward chaining, the same example is used by using the forward chaining method instead of the backward chaining.

STEP I. Forward Chaining

Given: DO is less than 1.0 mg/L and
aeration basin surface is wavy

Inference Engine

Action 1: Identify the known facts

Action 2: Which rules are related to the known fact?

Knowledge Base

Rule 1

**IF DO \leq 1.0 mg/L
THEN DO Level= Low**

Rule 2

**IF Surface= Calm
THEN Turbulence= Normal**

Rule 3

**IF Surface= Wavy
THEN Turbulence= Violent**

Rule 4

**IF Turbulence= Violent
AND DO Level= High
THEN Blower Rate= Mid**

Rule 5

**IF Turbulence= Violent
AND DO Level= Low
THEN Blower Rate= High
Advice "Repair diffusers
immediately"**

In forward chaining, the first step is to identify the facts which are given. In this case, the user tells expert systems that "DO \leq 1.0 mg/L" and "Surface= Wavy." Once the facts are known, rules are analyzed in a sequential manner. RULE 1 has a premise which is related to the known fact. The inference engine evaluates the premise and finds out that it is true. The conclusion is then executed and a value of "Low" for "DO Level" is assigned. The inference engine continues with the second rule. Again the rule's premise is related to the known fact so the engine fires the rule. The premise fails because the value of "Surface" is not "Calm." The inference engine then skips to the next rule. RULE 3 is satisfied and the value of "Violent" for "Turbulence" is assigned.

STEP II. Forward Chaining

Given: DO is less than 1.0 mg/L and aeration basin surface is wavy

Inference Engine

Action 1: Identify the known facts

Action 2: Which rules are related to the known fact?

Action 3: Evaluate RULE 1

Action 4: Assign value to "DO Level"

Action 5: Check RULE 2

Action 6: Evaluate RULE 3

Action 7: Assign value to "Turbulence"

Action 8: Consider RULE 4

Knowledge Base

Rule 1

IF DO \leq 1.0 mg/L
THEN DO Level= Low

Rule 2

IF Surface= Calm
THEN Turbulence= Normal

Rule 3

IF Surface= Wavy
THEN Turbulence= Violent

Rule 4

IF Turbulence= Violent
AND DO Level= High
THEN Blower Rate= Mid

Rule 5

IF Turbulence= Violent
AND DO Level= Low
THEN Blower Rate= High
Advice "Repair diffusers
immediately"

The inference engine now has a value for both "Turbulence" and "DO Level." The evaluation of RULE 4 proceeds. Two premises of RULE 4 are checked. The first one is true and the second one is false so the overall value is false. Thus, RULE 4 fails. The next rule is then analyzed. RULE 5 checks out true and the conclusion of the rule is executed. The same action is presented to the user.

The major difference between forward and backward chaining is that forward chaining checks the rule's premise, not the conclusion. Once the premise is satisfied, forward chaining executes the rule immediately regardless of the contents of the conclusion. Forward chaining also differs in other respects. Forward chaining goes through the rules in the order in which they are written in the knowledge base, independent of both premise and conclusion. This factor plays an important role in designing the structure of the knowledge base particularly for determining the sequence of rules within the knowledge base.

The third difference between forward and backward chaining is that forward chaining does not work on only one rule until it determines whether it has "passed" or "failed." It goes through the rules regardless of knowing all the variables in the rule. If any variable is set to NOTSOUGHT, forward chaining puts that rule aside and continues with the next rule. When the value of a specific variable is found, forward chaining goes back to the rule and evaluates it.

Mixed chaining is a combination of both forward and backward chaining. The method uses advantages from both algorithms. In some cases, mixed chaining is the most suitable application for obtaining the maximum performance from expert systems. Many mixed chaining techniques first start with backward chaining. Once a new value is assigned to any variable, the search algorithm is switched to forward chaining. Forward chaining goes through the rules and picks up the rules for the new known fact. Once the procedure passes through the knowledge base, backward

chaining is resumed. Again when a new value is assigned, forward chaining is initiated. All the rules are analyzed by both backward and forward techniques. Mixed chaining eliminates the disadvantages of backward chaining by introducing forward chaining. The example on the control of the blower rate can be used to illustrate the effect of mixed chaining on the search strategy.

STEP I. Mixed Chaining

Given: DO is less than 1.0 mg/L and
aeration basin surface is wavy

Find GOAL: Blower Rate

Inference Engine

Current GOAL: Find Blower Rate

Action 1: Which rules conclude the current goal?

Knowledge Base

Rule 1

IF $DO \leq 1.0$ mg/L
THEN DO Level= Low

Rule 2

IF Surface= Calm
THEN Turbulence= Normal

Rule 3

IF Surface= Wavy
THEN Turbulence= Violent

Rule 4

IF Turbulence= Violent
AND DO Level= High
THEN Blower Rate= Mid

Rule 5

IF Turbulence= Violent
AND DO Level= Low
THEN Blower Rate= High
Advice "Repair diffusers
immediately"

Similar to backward chaining, mixed chaining first searches for the rules which conclude the current goal. After RULE 4 is found, the inference engine checks the rule's premise block. The variable has an attribute of NOTSOUGHT. Backward chaining now changes the current goal to find "Turbulence." The rules concluding the variable, RULE 2 and RULE 3, are analyzed and it is found that RULE 2 is failed and RULE 3 is passed. Once RULE 3 is passed, the new value of "Violent" is assigned to "Turbulence." Now the inference engine switches to forward chaining.

STEP II. Mixed Chaining

Inference Engine

- Action 1: Which rules conclude the current goal?
- Action 2: Find the value of Turbulence
- Action 3: Which rules conclude "Turbulence"?
- Action 4: Evaluate RULE 2
- Action 5: Check premise in RULE 4
- Action 6: Search for another rule which concludes current goal
- Action 7: Evaluate RULE 4
- Action 8: Assign value to "Turbulence"
- Action 9: Switch to forward chaining

Knowledge Base

Rule 1

**IF DO \leq 1.0 mg/L
THEN DO Level= Low**

Rule 2

**IF Surface= Calm
THEN Turbulence= Normal**

Rule 3

**IF Surface= Wavy
THEN Turbulence= Violent**

Rule 4

**IF Turbulence= Violent
AND DO Level= High
THEN Blower Rate= Mid**

Rule 5

**IF Turbulence= Violent
AND DO Level= Low
THEN Blower Rate= High
Advice "Repair diffusers
immediately"**

Because forward chaining looks at the rule's premise, RULE 4 and RULE 5 are picked up. Forward chaining tries to satisfy these rules' premises but it fails because the "DO Level" attribute is set to NOTSOUGHT. Now, backward chaining is called back to search for the value of "DO Level." Once backward chaining assigns a value to "DO Level" as explained earlier, forward chaining is called back on the job. Again forward chaining searches the rules' premises and identifies RULE 4 and RULE 5. This time all the premises are known and RULE 5 is passed. The same solution is obtained.

In this section, the different search algorithms, backward and forward techniques are analyzed. It is important to apply the right search algorithm. The type of search algorithm may affect the performance of the expert system application which has a large knowledge base (Geselbracht and Johnston, 1988). The most suitable one, however,

depends on the type of the expert system application. Each method has advantages and disadvantages. A few things to consider in applying the right search algorithm are shown in Table 2-17.

Table 2-17: Questions relevant to choosing forward and backward chaining (Pedersen, EXPERT SYSTEMS PROGRAMMING, ©1989, p.80. Reprinted by permission of John Wiley & Sons, Inc.)

Trait	Forward	Backward
Solutions are:	not preenumerated	preenumerated
The goal is:	not necessarily known	known
The object is to:	flush out all facts	infer one key fact
Starting situation:	some facts known	few facts known
"Question and answer" type input required from user:	no	yes
Percent of rules typically applying to one case:	relatively high	relatively low
Strategy:	build solution	detect solution
Problem type (assuming rules are appropriate):	configuration planning interpretation	classification selection diagnostic

2.2.2.7 Uncertainty in the Rules

When an expert person uses "gut feelings" during the decision making process, the answer contains uncertainty. The uncertainty reflects the answer's credibility. The expert person must declare the reliability of the given information. The reliability is reflected to the expert systems knowledge base rules with a *confidence factor*, or also referred as *certainty factor* (CF).

The confidence factor reflects the degree of confidence the expert person has in the knowledge. Scales such as 0 to 5, -100 to 100, or most commonly 0 to 100 are used to quantify confidence factors. In a 0 to 100 scale, 0 indicates that the expert person has no confidence in the decision (total unreliability) and 100 is used to show a complete trust in the answer (total credibility). For example, the expert person says:

"When there is a violent turbulence in the aeration basins, the dissolved oxygen concentration will *likely* be more than 3.0 mg/L."

Words, such as "sometimes," "likely," "slight change," and "maybe," indicate the person's reliance on the knowledge. Preassigned values for such words may be included in the knowledge acquisition process and carried out to the development of the knowledge base. The above example can be formulated in a rule form as follows:

IF Turbulence = Violent
THEN DO > 3.0 mg/L **CF 75**

In the above example, a 75% value is assigned to the word "likely." Confidence factors can be in both the premise and the conclusion blocks. If the existence of a condition in the premise is not certain, then a CF can be assigned to that block. Similarly, if the outcome of a rule is doubtful, the conclusion statement carries a confidence factor less than 100 to reflect some unreliability. If there is no CF in a rule, it is assumed to be 100% certain.

One important property of confidence factors is the truth threshold. The truth threshold is a predetermined confidence factor value which is used to verify if the premise is true or false. If the premise CF has a higher value than the truth threshold, the premise is considered "true" and the rule passes. However, having a lower CF value than the truth threshold will cause the rule to fail.

The truth threshold is only applied to the premise block and not the conclusion block because a conclusion may be doubtful but still exist. For instance, if the truth threshold is set to 95, the following rule would fail because the premise's CF is lower than the truth threshold ($90 < 95$).

IF DO < 1.0 mg/L **CF 90**
THEN Problem = Clogged diffusers **CF 60**

In the process of expert system consultation, the confidence factors can be changed. CFs can either be preassigned within the knowledge base or the user may assign CFs during the consultation sessions. Both confidence

factors for the premise and the conclusion blocks make up the rule's combined CF.

Similarly, if more than one rule is used to achieve a solution, the CF of the solution is determined by manipulation of all the rules' combined CFs. The final combined CF directly reflects the reliability of the final solution. A sample calculation of a rule's combined CF is illustrated by the following example.

IF DO < 1.0 mg/L **CF 90**
THEN Diffusers = Clogged **CF 60**

The above example rule shows that there is a 10% error (90% confidence) in measuring the DO when the DO is below 1.0 mg/L, and if the DO reads below 1.0 mg/L, then diffusers might (60% chance) be clogged. Assuming the rule's premise is true (*i.e.*, the DO is below 1.0 mg/L), the rule's overall CF is calculated by the following equation (Note that adding 50 to the denominator will round the number to the next higher fraction, Texas Instruments, Inc., 1988):

$$CF (rule) = \frac{\text{premise CF} \times \text{conclusion CF} + 50}{100} \quad (2-41)$$

$$CF (rule) = \frac{90 \times 60 + 50}{100} = 55 \quad (\text{Rounded}) \quad (2-42)$$

where $CF(\text{rule}) = \text{Rule's combined CF}$
premise $CF = \text{CF of the } IF \text{ block}$
conclusion $CF = \text{CF of the } THEN \text{ block}$

It is also possible that rule may have more than one premise or conclusion which each has different confidence factors. When an *IF* (...) block is conjugated by an "AND" word, the overall premise CF is equal to the *lowest* CF within that *IF* (...) block. For example, the following rule has multiple CFs as shown below:

<i>IF</i>	Turbulence = Violent	CF 85
<i>AND</i>	DO < 1.0 mg/L	CF 95
<i>THEN</i>	Problem = Clogged diffusers	CF 60

The above sample rule indicates that if turbulence is violent and the DO reading is below 1.0 mg/L, then the problem could be caused by clogged diffusers. The example above contains multiple premise CFs and a conclusion CF. Assuming the rule premise is true, the overall CF--rule's combined confidence factor--can be calculated by first determining the premise CF and then combining that CF with the conclusion CF.

When the *lowest*-CF-method is applied in the above example, the combined premise CF becomes 85 because the first premise's CF is less than the second premise's CF ($85 < 95$). Once the combined CF for the premise block is determined, the rule's combined CF can be calculated by using Eq. 2-41. The multiple CFs will, therefore, yield the following CF:

$$CF (rule) = \frac{(85) \times 60 + 50}{100} = 52 \quad (Rounded) \quad (2-43)$$

If the "OR" operator joins the premises within the *IF* (...) block, the combined CF for the overall premise is calculated based on the *highest*-CF-method which assigns the highest CF among the premises to the combined premise CF. For example, the previous sample rule is modified as follows:

IF Turbulence = Violent CF 85
OR DO < 1.0 mg/L CF 95
THEN Problem = Clogged diffusers CF 60

By applying the *highest*-CF-method, the combined premise CF in the above rule becomes 95. The rule's combined CF is then calculated by Eq. 2-41 as follows:

$$CF (rule) = \frac{(95) \times 60 + 50}{100} = 58 \quad (Rounded) \quad (2-44)$$

A combination of the "AND" and the "OR" functions may exist. In these cases, first the combined CF for the "AND" premises are determined by using the *lowest*-CF-method. Then, this product CF is treated along with the rest of the "OR" premises by using the *highest*-CF-method.

During the consultation, the inference engine keeps track of the CFs for each parameter and the rules used. If several rules conclude the same parameter and each rule assigns different confidence factors to the same conclusion, the final CF will be affected by all CFs which are involved. If a parameter has been evaluated prior to the new rule's testing, the parameter has a previous CF value which is symbolized as CF(previous). For example, after passing the first rule, the parameter A has CF 55. For the examination of the second rule, the parameter A's CF becomes its CF(previous) 55. The following equations show how the CF(previous) and the CF(rule) values are associated. The equations are adapted from the Personal Consultant™ Plus user manual (Texas Instruments, Inc., 1988).

If $CF(\text{previous}) \geq 0$ and $CF(\text{rule}) \geq 0$, use the following equation:

$$CF = CF(\text{previous}) + \frac{CF(\text{rule}) \times (100 - CF(\text{previous})) + 50}{100} \quad (2-45)$$

If $CF(\text{previous}) < 0$ and $CF(\text{rule}) < 0$, use the following equation:

$$CF = CF(\text{previous}) + \frac{CF(\text{rule}) \times (100 + CF(\text{previous})) - 50}{100} \quad (2-46)$$

If $((CF(\text{previous}) \times CF(\text{rule})) < 0$ and $(CF(\text{previous}) + CF(\text{rule})) > 0$), use the following equation:

$$CF = \frac{(CF(previous) + CF(rule)) \times 100 + \frac{100 - MIN}{2}}{100 - MIN} \quad (2-47)$$

If ((CF(previous) x CF(rule)) < 0 and (CF(previous) + CF(rule)) < 0), use the following equation:

$$CF = \frac{(CF(previous) + CF(rule)) \times 100 - \frac{100 - MIN}{2}}{100 - MIN} \quad (2-48)$$

where MIN = CF(previous) or CF(rule) whichever has lesser absolute value.

To simplify the use of CF(previous) and the combined CFs, the previous example will be expanded. The following facts are given:

- The parameter "Problem" has been evaluated and has a CF(previous) 52 (see Eq. 2-43).
- The following rule also concludes the parameter "Problem."

<i>IF</i>	Blower Pressure	> 20 psi	CF 100
<i>AND</i>	Organic Loading	< 500 kg/d	CF 80
<i>AND</i>	MLSS	> 3000	CF 90
<i>AND</i>	Sludge Age	> 20 d	CF 75
<i>THEN</i>	Problem = Clogged Diffusers		CF 80

Calculation of the final CF is outlined step by step in the following section.

Step 1: Because the example rule's premises are combined with the "AND" function, the *lowest-CF-method* is used to determine the combined premise CF. The lowest CF within the premises is 75.

Step 2: Calculate rule's combined CF using Eq. 2-41.

$$CF(\text{rule}) = \frac{75 \times 80 + 50}{100} = 61 \quad (\text{Rounded}) \quad (2-49)$$

Step 3: Since both the CF(previous) and the CF(rule) are positive, the Eq. 2-45 is used to determine the final CF for the parameter "Problem."

$$CF = 52 + \frac{61 \times (100 - 52) + 50}{100} = 82 \quad (\text{Rounded}) \quad (2-50)$$

The confidence factor for the parameter "Problem" has increased because there is more than one rule which indicates the actual "Problem" is "Clogged Diffusers." Similarly, the parameter's CF increases with the increasing number of higher CFs. However, if there is conflicting evidence (lower CF) of the "Problem," then the original CF value for the "Clogged Diffusers" will decrease to indicate the increasing unreliability.

Assigning confidence factors to the rules and the parameters is a significant step in the development of a knowledge base. There are four

major methods to assign CF values to the variables and the rules. These methods are briefly explained below:

- ***Default CFs:*** During the knowledge base development, the developer enters the default CFs for the parameters which carry some uncertainty. For instance, the dissolved oxygen parameter has a default CF 90 because there is a 10% error measuring the oxygen level. The default CFs are determined based on the equipment limitations and the credibility of the information.
- ***Pre-assignment of the conclusion CFs:*** In addition to the parameters having CFs, rules also may display some uncertainty originating either from the expert person's expressions or from the physical laws. The rule's conclusion block contains the pre-assigned CFs. In the earlier example, the rule which concludes the parameter "Problem" had a CF of 70. This indicates that the conclusion statement has a 70% credibility if the premise block passes. At the end of the consultation, the user is informed about the 70% reliability.
- ***CF assignment by user response:*** During the consultation session, a number of questions are asked of the user. The user may indicate uncertainty about his/her answers. For example, the expert system asks the following question: "Do you expect to receive rain-flow into the treatment plant within the next 24-hours?" and the user responds: "Yes CF 60." The user indicates that there is a 60% chance of rain and the expert system takes preventive measures to compensate for the increasing flowrate due to the rain fall.
- ***Determine CF by real-time data bases:*** Use of *empirical statistic* methods to assign the CFs is also an option for the expert system developers. The statistical methods calculate probabilities based on "real-time frequencies." One of the best known techniques in empirical statistics is the *Bayesian statistics* approach. In Bayesian statistical model, the real-time frequency of a variable is empirically calculated to assign a confidence factor for that variable. More detailed analysis of the Bayesian statistics are discussed in section 2.2.3.

The use of confidence factors in the development of a knowledge base is unavoidable. Describing heuristic knowledge is made much more realistic by assigning CF values to uncertain rules. The CFs provide a vital link between the expert systems and the real-time applications.

2.2.2.8 Knowledge Base Development and Organization

The development and organization of a knowledge base has a significant impact on the performance of the expert systems. In this section, the knowledge base creation and the organization techniques are analyzed.

An effective knowledge base is the one which uses all the advantages that expert systems programming offers. The most important issues in the development of a knowledge base are: (1) clear-cut representation of knowledge by the rules; (2) organization of these rules; (3) simple structure; and (4) being designed for a specific type of inference engine usage. If all these factors are compiled, the knowledge base will provide a high performance output.

During formulation of the rules, clear and simple words should be used so that what the rule represents will be easily understood. Descriptive, English-like names are preferred to names only the developer is familiar with. Simple writing of the rules is an important factor because once the untrained user is left alone with the expert systems, he/she should be able

to update and maintain the rules within the knowledge base without a high level of computer expertise.

Organization of rules within the knowledge base may significantly improve both the performance and the consultation time of an expert system. The knowledge base can be integrated into expert systems structure as shown in Figure 2-16. The internal structure of the knowledge base should be simple so that any user can go into the knowledge base and modify rules for their specific applications.

Location of a rule, in fact, does not affect whether the rule is going to be processed or not. Performance and speed of the expert system, however, are greatly influenced by the way the rules are placed within the knowledge base. The more rules in the knowledge base, the more important the organization becomes (Geselbracht and Johnston, 1988).

One method of organization is to categorize rules by subject in frames. A frame is the smallest part of a knowledge base and contains a series of rules which are related to each other. Each knowledge base must have at least one frame--the root frame.

Several different groups of rules may exist within the same knowledge base structure. The root frame in a complex knowledge base contains additional frames called sub-frames. For example, the root frame may contain all general wastewater treatment plant process control rules and the specific rules about wastewater influent characteristics may be kept under

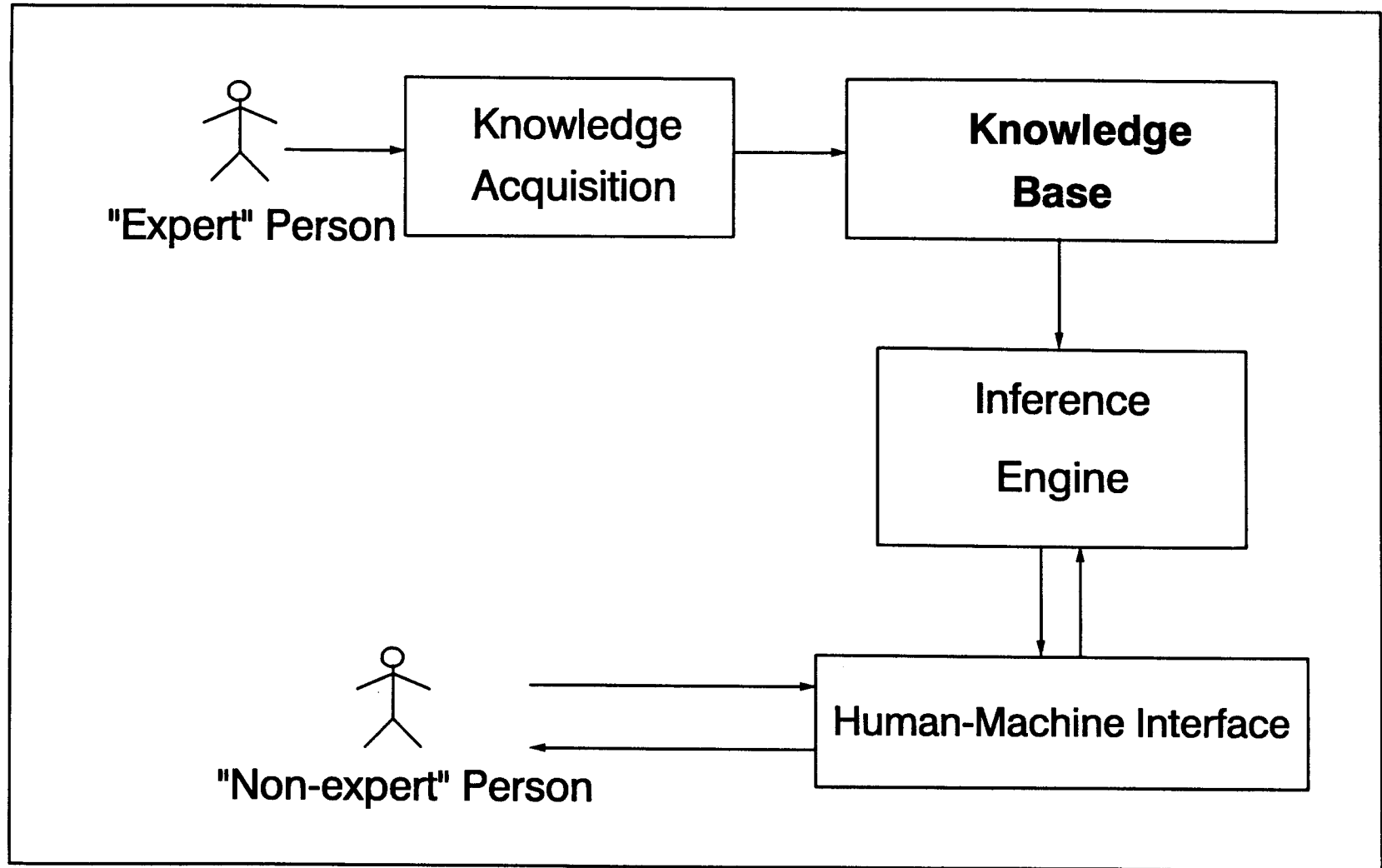


Figure 2-16: Knowledge base component within the expert system

the "Waste influent" sub-frame, and aeration tank rules are in the "Aeration" sub-frame.

Each sub-frame is considered the child frame of the previous frame. Similarly, a frame which has a child frame is called a parent frame. Certain information is exchanged within the child and the parent frames through inheritance. Although the child frame inherits the parameter values from its parent frame, the parent frame does not have access to the child frame's parameters. The child, the parent, and the inheritance concepts can be illustrated schematically as shown in Figure 2-17.

In Figure 2-17, the root frame has two branched sub-frames. The sub-frames A and B are the child frames of the root frame. Therefore, the root frame is the parent frame to the sub-frames A and B. Similarly, the sub-frame B has two child frames, the sub-frames C and D. Overall, the sub-frame C and D have two parents, the sub-frame B and the root frame.

Inheritance plays an important role in the parameter exchange process. The sub-frames C and D can access the parameters in sub-frame B and the root frame. But the parameters in sub-frame A are not accessible by the rules in sub-frames C and D. Sub-frame B also does not have access to its child frames' parameters. During the organization of the sub-frames, the inheritance factor should be taken into account to deplete any disturbances in the flow of the parameter exchange procedure.

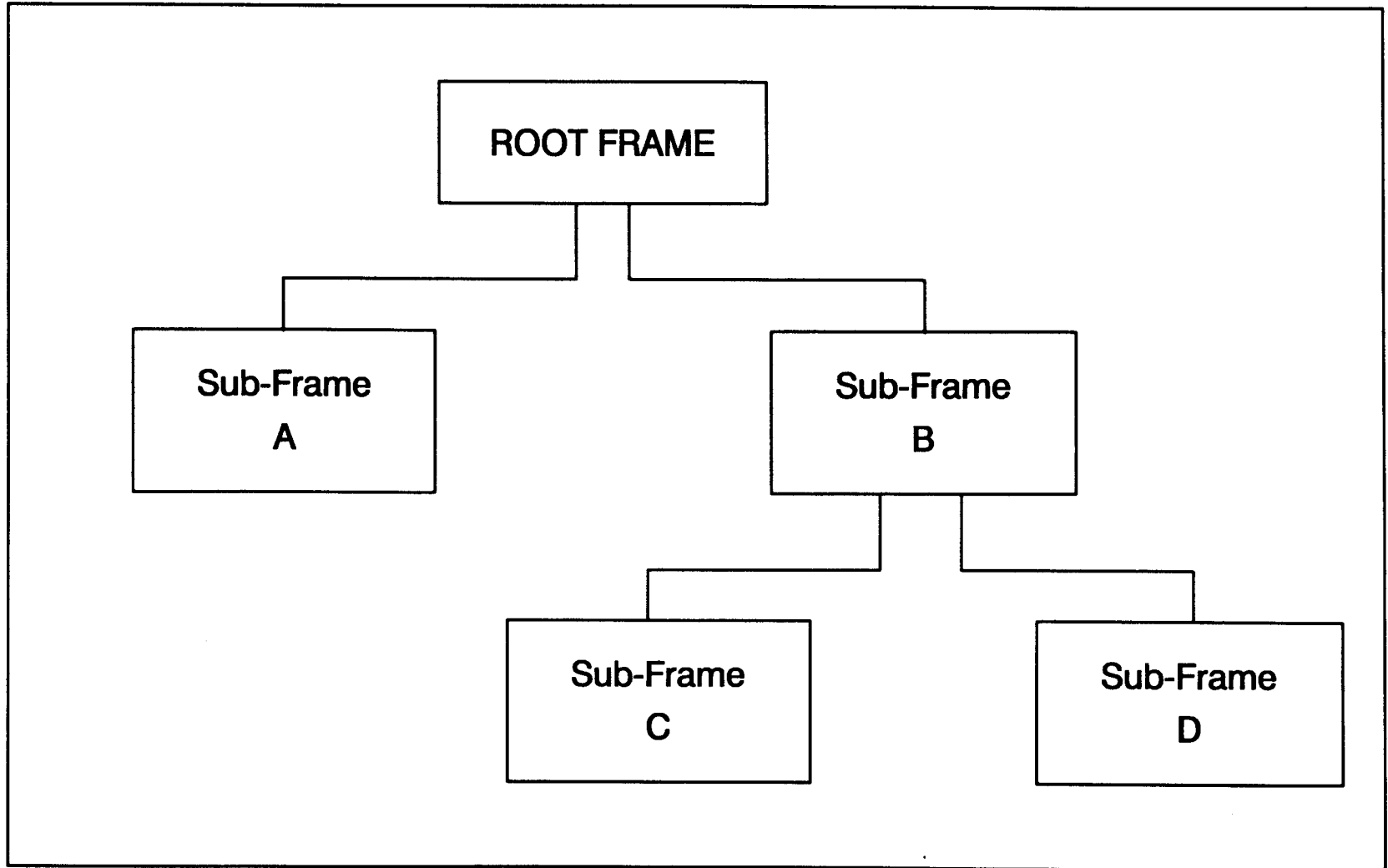


Figure 2-17: Frame structure of knowledge base

Internal organization of a sub-frame also has an effect on the computational speed. The rules most likely to be used should be placed at the beginning of each sub-frame. In other words, the rule which provides the most frequent conclusion should appear first. For example, RULE 1 is placed at the beginning of the sub-frame because it concludes the most frequent "GOAL." RULE 1 contains several premises and some of them have an attribute value of NOTSOUGHT. Because the inference engine cannot conclude this rule without all the premises of the rule being satisfied, the inference engine is forced to locate the rules which conclude the premises of RULE 1. These premises are named "1° premises" because they are the first ones to appear in the search process. RULE 2 concludes the 1° premises and it is placed just after RULE 1. The placement of additional rules are necessary because RULE 2 has premises which are needed to be evaluated. So RULE 3 which contains conclusions for RULE 2 premises (2° premises) is placed next and additional rules follow until no more premises are left to evaluate.

In the above example, backward chaining finds the rules quickly and reaches a solution faster. Because the inference engine stops as soon as the goal is satisfied, the time spent during the search procedure is improved. This type of organization is efficient, even when there are multiple goals within the knowledge base. The multiple goals are arranged separately into different sub-frames and organized individually within its own sub-frame.

The above mentioned technique provides benefits to both the inference engine speed and the consultation time. Because less switching between the sub-frames is required and the time spent during the search procedure is reduced, the inference engine gains significant speed. Furthermore, the user can read, modify, or update rules easily because all related rules are located in the same place.

Although all the precautions can be taken, several problems may exist in a knowledge base. These problems can be overcome by the debugging and testing procedures discussed in the next section.

2.2.2.9 Knowledge Base Testing

There is no accurate technique to either capture or represent knowledge without impurities (Gruber, 1989). Integrity of a knowledge base directly affects the validity of an expert system consultation. Any discrepancy in the logic results in misleading output. In other words, the expert system advises the user erroneously.

The problems with the knowledge bases which are associated with misrepresentation of the knowledge are the most difficult ones to debug. Because the computer does not recognize "wrong" knowledge, it is difficult to locate the problem area. The computer believes whatever its developer tells it to believe. This trust, however, causes serious defects in expert systems functions.

To avoid possible errors, first the developer should pick the most suitable knowledge representation technique to apply to that specific use. Second, the developer should study the chosen representation technique structure in depth to identify the pitfalls. The following problems are the most frequently observed ones in the knowledge representation.

- There might be more than one rule which has the same premise block, but one or more rules may conclude the same attribute differently--conflicting rules.
- The "gaps" in knowledge, which can be defined as missing rules within the knowledge base, cause serious problems such as unavailability of a value for an attribute.
- The knowledge might be organized incorrectly within the logic trees.

All the above knowledge representation problems result in failure of the expert systems.

Other causes for the failure include: the use of too specific or complex rules; inadequately named attributes; and rigid rules. If a rule is written in a complex form such as multiple *IF* (...) blocks within the premise block, it becomes harder to understand and to maintain. The user gets lost within the rule trying to figure out how to modify the rule.

If a rule contains several "AND" conditions, the rule might be too specific. Thus, it may never be used by expert systems because such a specific condition exists to infrequently. The number of "AND"s to use

should be carefully analyzed and if it is necessary, multiple rules should be created to achieve "generality."

Rules which have illegible names are more likely to cause problems because the user cannot figure out to what the parameters are referring. The rules should be designed for any user who wishes to use the expert systems. The use of descriptive names for rules will improve the quality of the knowledge base.

Finally, inflexible rules are the main reason in the failure to expand knowledge bases. Many inflexible rules contain numbers and these numbers are assigned to the objects directly. If the value assignment is performed in a "hard way" which is a "DO= 1.0 mg/L" like assignment, it is difficult to update the parameter. The user has to go over every rule in the knowledge base to find similar values and change these manually. This procedure is frustrating and time consuming. Instead, new attributes such as "low" or "high" should be introduced to make the rule flexible. For example, one rule says "DO= Low," and the other rule defines the "low" as "Low= 1.0 mg/L." This way the user has a chance to update the entire "low" attribute by changing only one rule. Or, the value for "low" can be assigned through text file import procedures which completely eliminate the need for altering the rules.

The integrity of a knowledge base is established by the knowledge base testing procedure. The testing procedure is performed by close observation of rule handling during the expert system consultation. The

knowledge engineer works along with the inference engine to find the bugs. Simply stated, the engineer enters hypothetical conditions and evaluates the results. Although testing all the possibilities is desired--so that all the rules are used, it is a time consuming task. Furthermore, in large knowledge bases, this option might be impossible to execute because of the large number of possibilities.

In order to perform proper testing on a knowledge base, a few requirements must be satisfied. First, the knowledge base should have an adequate number of rules (*i.e.*, there are enough rules present to conclude several conditions). Early testing in a large knowledge base is a good idea because a large number of rules are difficult to evaluate at the same time. Secondly, testing should be performed by an expert person or by an experienced developer, in order to evaluate the accuracy of the advice given by the expert system. Finally, the last stage of testing should be accomplished by applying expert systems to the real-time operations. An example of real-time testing of the expert system for the activated sludge process is given in Lai and Berthouex (1990).

2.2.2.10 Data Base Structure

The next step in developing expert systems is to couple knowledge base with a data base. The data base is incorporated into the expert systems' structure as shown in Figure 2-18. The most important task of the data base component is to generate the confidence factors which are specific to that treatment plant.

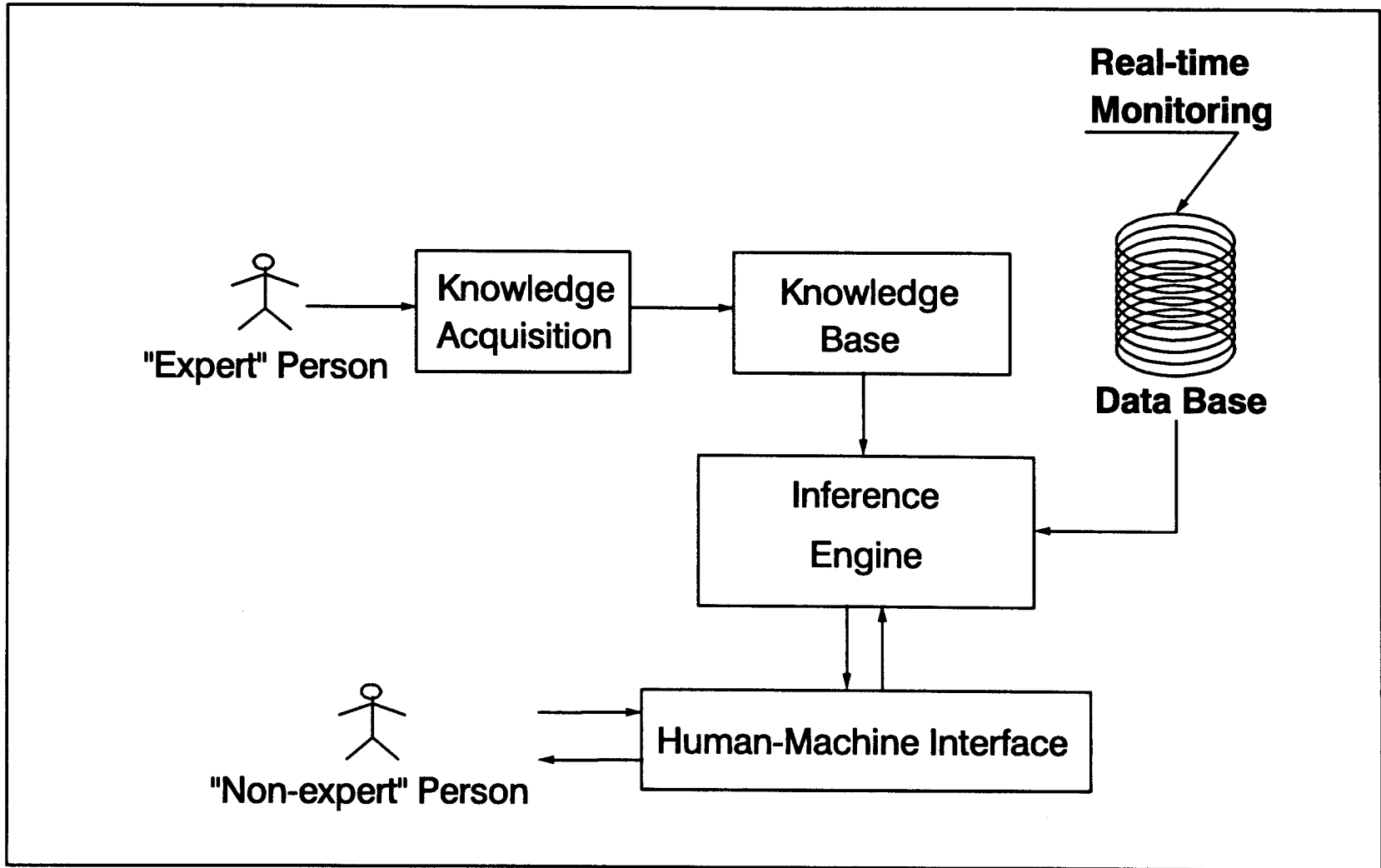


Figure 2-18: Data base component within expert system

In data bases, both the quantity and the quality of data have an effect on the statistical calculations. The quantity of data is important because statistical method calculates CFs based on the frequency of appearance for the specific condition. In other terms, the more the condition is observed in the past, the better representation of the system is obtained with the statistical methods. In order to establish a reliable sample group for the statistical calculations, a large amount of data should be available. This may create some problems, particularly, if the treatment plant is new or the plant has not practiced historical data collection.

Besides the quantity, the quality of data is also an important factor. Reliability of the confidence factor estimations by the statistical models is directly related to how much the data represents the truth. A problem-free data set is difficult to obtain. There are always limitations in the process control instruments which results in some kind of error value. More reliable data may be supplied by computer-assisted systems (Ostgaard *et al.*, 1985). However, the data obtained from real-time applications always reflects an error value, regardless of the way the data is processed. Because there will always be error in the data sets, this fact discourages most expert system developers from using the statistical models.

Many researchers have worked on the data base development techniques. The way to create an efficient, reliable, manageable data base is outlined in the literature. Analysis of the real-time data has received considerable attention (Hao and Neethling, 1987; Jayawardena and Lai, 1989). Uncertainty analysis and the data base applications for the expert systems

have also been researched (Blockley, 1987; Flanagan and Finnan, 1987; Berthouex *et al.*, 1989; Parker *et al.*, 1989).

The basic function of a data base is illustrated in Figure 2-19. In Figure 2-19, the equipment measures the real-time values and then sends this information to the data base. The inference engine accesses the data base and calculates the confidence factors for the rules which are found during the expert system consultation. During the consultation, the inference engine consults the data base several times to update the current CFs. Additional support from the statistical methods enhances the quality of expert system's advice. As a result, a link between the real-time values and the expert system is established by the data base component.

The data base may provide more reliable values to the decision making process performed by the inference engine. Since most of the wastewater treatment plants collect extensive amounts of data, this dissertation will take advantage of such existing data bases to improve the expert systems' functions.

2.2.3 Statistical Models: Bayesian

The importance of the confidence factors in the rules is explained in the previous sections. This section will investigate how the confidence factors are calculated by using the historical real-time values of a specific condition within the data base.

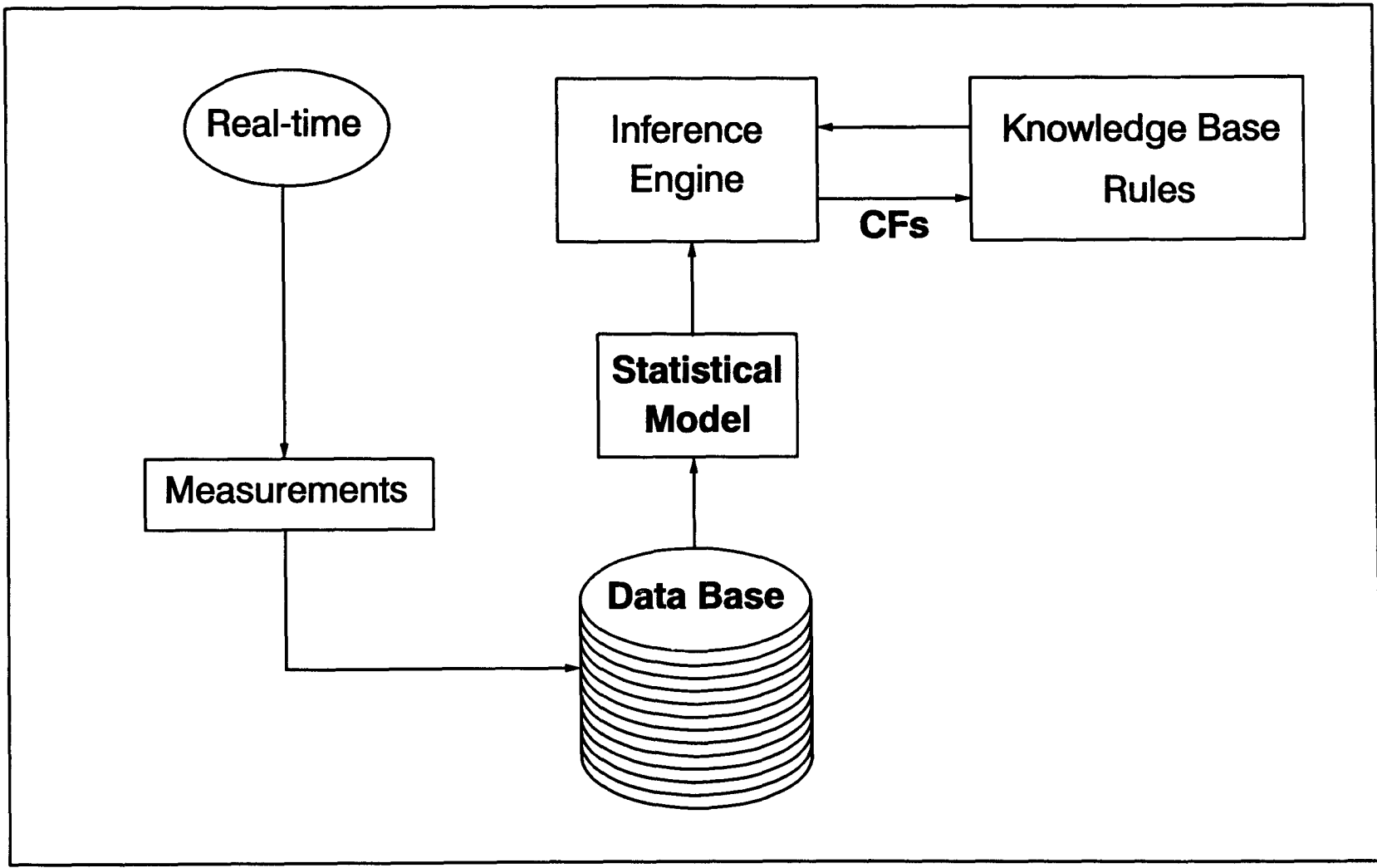


Figure 2-19: Data base functions during expert system consultation

A number of occurrences of a specific condition, which is called the *frequency analysis* in the statistical methods' terminology, are used to estimate the confidence factors. However, the frequency analysis alone is not an adequate tool for assigning the confidence factors. The second generation of the statistical models, such as *Bayesian statistics*, provide more realistic uncertainty measurements--CFs.

A quick, rough estimate on the likelihood the condition will occur can be obtained by the frequency analysis. For example, assume there are n number of data points available in the historical data base. Within that n number of data, a specific condition, which is called *Condition A*, is observed x times. Frequency of *Condition A* can be calculated with the following formula (Spiegel, 1961):

$$\text{Frequency of Condition A} = \frac{x}{n} \quad (2-51)$$

where x = Number of times *Condition A* is observed in the past
 n = Total number of events observed in the past

The frequency of a specific condition, therefore, can be defined as a fraction of that specific condition existed within the events which have occurred in the past.

One property of the frequency should be emphasized here. The frequency value shows what happened in the past. This is a known fact and it is not an estimation. Probability, in contrast, is a measurement of likelihood or an estimation. Although, the frequency values are used to interpret the

future likelihood, the frequency values demonstrate only the past not the future. In many cases, however, frequency values are used to calculate probability of having a specific condition in the future based on the assumption that the "past will repeat itself."

The issue of predicting the future has been a focus area in the statistics field for a long time. Early discoveries of the frequency analysis, namely the Bayesian statistics, created a new approach for calculating uncertainty. In order to explain the Bayesian statistics, two definitions are necessary. First, *Sample* is used to describe the parameters which can be measured or controlled. Sample could be anything which can be adjusted by a speed controller or by an ON/OFF switch such as the blower speed, or the influent valve open/close switch. Sample variables are not limited to the above examples. Anything that can be measured may also be considered sample such as wastewater influent characteristics, the DO in the tanks or temperature value.

Every sample causes a condition to happen. These conditions, which are attributable *only* to the sample variables, appear in the real-time applications and are referred to as *States*. State, in other words, is a condition which is caused by an existing sample in the past. Figure 2-20 illustrates how a sample and a state are related to each other. In Figure 2-20, $Sample_i$ caused $State_j$ in the past and $Sample_{i+1}$ will cause $State_{j+1}$ with some degree of uncertainty in the future. Some examples for state include: effluent water quality, and the DO level or turbulence in the aeration basins.

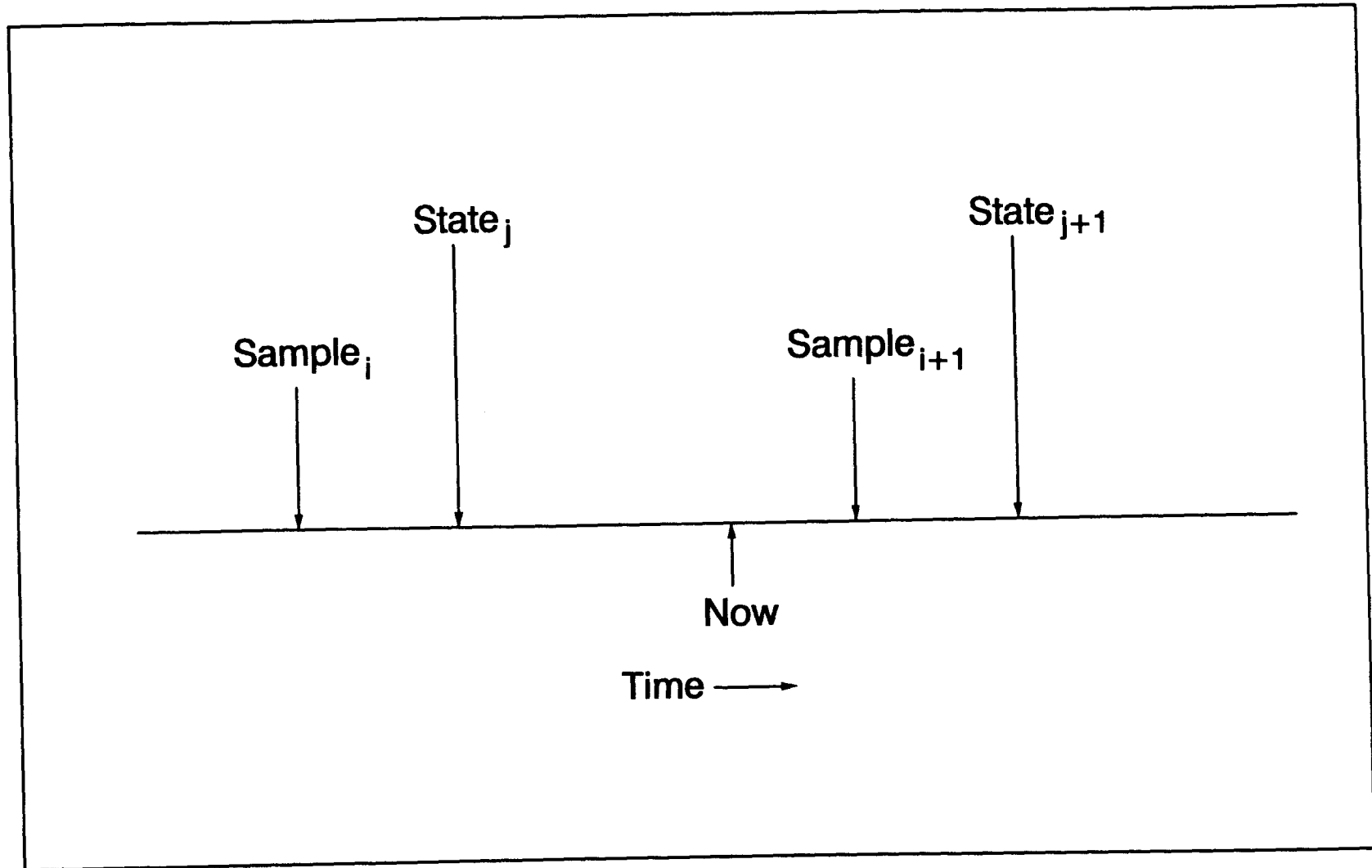


Figure 2-20: Definition of *State* and *Sample* variables

The Bayes theorem (*i.e.*, Bayesian statistics) states that the likelihood of having a specific state--condition--under given specific sample--evidence--is a function of both prior and conditional probabilities. **Prior** probability refers to the frequency of such a specific state occurring in the past, regardless of the sample's influence. **Conditional** probability can be defined as the frequency of a specific sample as it existed when a given state occurred for that time period. **Posterior** probability is the probability of the future happening. A posterior probability for having a certain *State_j*, when there is certain *Sample*, can be formulated by the Bayesian statistics as shown below:

$$P [State_j | Sample] = \frac{P [Sample | State_j] \times P [State_j]}{P [Sample]} \quad (2-52)$$

where

$P [State_j | Sample]$ = is conditional probability of having *State_j*, when *Sample* exists (posterior probability).

$P [Sample | State_j]$ = is conditional probability of observing *Sample* when *State_j*, has occurred (conditional probability).

$P [State_j]$ = is prior probability of *State_j*, regardless of whether *Sample* exists or not (prior probability).

$P [Sample]$ = is prior probability of *Sample* regardless of whether *State_j*, has occurred or not (prior probability).

If there is more than one state for a given sample, the Bayesian statistics can be written as follows:

$$P [State_j | Sample] = \frac{P [Sample | State_j] \times P [State_j]}{\sum_{i=1}^m (P [Sample | State_i] \times P [State_i])} \quad (2-53)$$

where

$P [Sample | State_i]$ = is conditional probability of *Sample* for specific *State_i* (all the states which are observed in the past, $i= 1, 2, \dots, m$).

$P [State_i]$ = is prior probability of *State_i* (all the states which are observed in the past) regardless of whether *Sample* exists (prior probability).

The above probability assumes there are "m" number of different states which have been observed for the same sample value in the past. For example, the sample value "High BOD loading" may have three different effects on the activated sludge process such as low DO in the basins (*State₁*), high BOD in the plant effluent (*State₂*), and increasing MLVSS in the basins (*State₃*). First, in order to calculate prior probabilities, the historical data base is needed. The example below illustrates use of the Bayesian statistics.

The example data base contains the following information:

Total Number of Data = 365

Dates Collected = year of 1989

Number of times which

DO in the basins was low = 132 times (State₁)

Out of 132 times, BOD loading was high = 73 times (Sample₁)

Out of 132 times, BOD loading was normal = 45 times (Sample₂)

Out of 132 times, BOD loading was low = 14 times (Sample₃)

Number of times which there was

high BOD in the plant effluent = 51 times (State₂)

Out of 51 times, BOD loading was high = 37 times (Sample₁)

Out of 51 times, BOD loading was normal = 11 times (Sample₂)

Out of 51 times, BOD loading was low = 3 times (Sample₃)

Number of times which

MLVSS increased in the basins: 215 times (State₃)

Out of 215 times, BOD loading was high = 176 times (Sample₁)

Out of 215 times, BOD loading was normal = 25 times (Sample₂)

Out of 215 times, BOD loading was low = 14 times (Sample₃)

Based on this information, prior probabilities can be calculated as follows:

$$P [\text{State}_1] = 132 \div 365 = 0.36$$

$$P [\text{State}_2] = 51 \div 365 = 0.14$$

$$P [\text{State}_3] = 215 \div 365 = 0.59$$

And conditional probabilities are calculated as follows:

$$P [\text{Sample}_1 | \text{State}_1] = 73 \div 132 = 0.55$$

$$P [\text{Sample}_2 | \text{State}_1] = 45 \div 132 = 0.34$$

$$P [\text{Sample}_3 | \text{State}_1] = 14 \div 132 = \underline{0.11}$$

1.00

$$\begin{aligned}
P [\text{Sample}_1 | \text{State}_2] &= 37 \div 51 = 0.73 \\
P [\text{Sample}_2 | \text{State}_2] &= 11 \div 51 = 0.22 \\
P [\text{Sample}_3 | \text{State}_2] &= 3 \div 51 = \underline{0.05} \\
&1.00
\end{aligned}$$

$$\begin{aligned}
P [\text{Sample}_1 | \text{State}_3] &= 176 \div 215 = 0.82 \\
P [\text{Sample}_2 | \text{State}_3] &= 25 \div 215 = 0.12 \\
P [\text{Sample}_3 | \text{State}_3] &= 14 \div 215 = \underline{0.06} \\
&1.00
\end{aligned}$$

Based on the prior frequency values, the Bayesian statistics can make a prediction for the future. For instance, there is a "High BOD loading" (Sample₁) in the wastewater influent. We want to know what the likelihood of having "low DO in the basins" (State₁). The Bayesian statistics calculates the likelihood of being in different states in the following form:

Sample₁ = "High BOD loading"
State₁ = "Low DO in the basins"

$$P [\text{State}_1 | \text{Sample}_1] = \frac{P [\text{Sample}_1 | \text{State}_1] \times P [\text{State}_1]}{\left(\begin{aligned} &P [\text{Sample}_1 | \text{State}_1] \times P [\text{State}_1] + \\ &P [\text{Sample}_1 | \text{State}_2] \times P [\text{State}_2] + \\ &P [\text{Sample}_1 | \text{State}_3] \times P [\text{State}_3] \end{aligned} \right)} \quad (2-54)$$

$$\begin{aligned}
P [\text{State}_1 | \text{Sample}_1] &= \frac{0.55 \times 0.36}{(0.55 \times 0.36 + 0.73 \times 0.14 + 0.82 \times 0.59)} \\
&= 0.25
\end{aligned} \quad (2-55)$$

In the above example, the probability of having low DO in the basins when there is a high BOD loading is found to be 25%.

Table 2-18: Example of the Bayesian statistical table

Bayesian Statistics Table	State ₁	State ₂	State ₃
Sample ₁	0.25	0.13	0.62
Sample ₂	0.55	0.14	0.31
Sample ₃	0.48	0.09	0.43

A Bayesian table similar to the one shown in Table 2-18 is created by calculating probabilities for various combinations of the sample and the state variables. The explanation of Table 2-18 is as follows: if a high BOD loading (Sample₁) is received in the plant, the most severe effect will likely be seen in the MLVSS increase (State₃) because the State₃ has the highest probability (0.62) when there is Sample₁ present. However, if the BOD loading is normal (Sample₂), there might be a chance that the DO level will drop in the basins (55% chance). The phenomena can be explained as the BOD loading does not have linear correlation with the DO level in basins for that particular treatment plant. If the BOD loading is low (Sample₃), there is very little chance (9%) the effluent BOD will increase. There is, however, a substantial chance there might be some problems with the DO and the MLVSS (48% and 43%, respectively).

Ability of using historical values to predict the future is not the only important property of the Bayesian statistics. The **updating** capability of

the Bayesian statistics is helpful to predict the future events by using readily available data (Brown, 1987; Kyburg, 1987). The method allows the prior probabilities to be used in the light of new information efficiently and it immediately reflects these changes into the posterior probabilities.

The prediction of the future by the Bayesian statistics can be reflected into the expert systems. While the inference engine is processing the knowledge base, confidence factors for the existing rules can be updated with the Bayesian statistics. As a result, expert systems can report accurate advice to the user with a likelihood attribute. The user, this way, will know how much to rely on the recommendation given by the expert system. Moreover, the Bayesian statistics may be used to refine the advice given by the expert system based on the past experiences with the process (Armijos *et al.*, 1990).

The Bayes theorem is sometimes avoided because of problems involved. The most common problem with Bayesian statistics is that there are not enough data points available to create statistically valid conditional frequency values which means a specific sample or state has not occurred frequently enough in the past (Kunz *et al.*, 1983). If there is one state which occurs every 10 years and we only have 9 years of data, the data base is not going to show any indication that such a state may occur. Eventually, the unexpected state occurs. This type of error is commonly seen in the Bayesian statistics. One way to detect such a problem is to handle probabilities differently which have a value of "0" (zero). It should

be noted that there is always some chance of having a certain state variable, even when the state variable repeats only every 100 years.

One way to cure problems with the Bayesian statistics is the use of *Subjective Bayesian* method (Duda *et al.*, 1976) where more realistic values for prior probabilities are assigned by the expert persons. The expert person enters prior probabilities as default values for the rules and the variables in the knowledge base. Probabilities directly reflect the degree of experience which that expert person possesses. Preassigned CF method, however, does not work well if it is used during the expert system consultation because if there is an expert person around during the consultation, the need for the expert system consultation becomes obsolete.

The second most common pitfall in the Bayesian statistics is that the method assumes *conditional independence* (*i.e.*, no relationship exists between observing State_x for a given sample and second State_z for the same sample value). Assuming the conditional independence may cause the Bayesian statistics to fail (Norusis and Jacquez, 1975). A possible remedy for this type of pitfall is to refrain from assuming conditional independence (Cumberpatch and Heaps, 1976). However, the solution becomes so complex that it causes other problems (see Shortliffe *et al.*, 1979).

There appears to be no easy way to use the Bayesian statistics. If the method is used under close supervision, it may benefit expert systems significantly. However, the Bayesian statistics may collapse anytime and cause the expert systems to advice chaotically.

2.2.4 Mathematical Models in Expert Systems

The expert systems are not solely built based on the heuristic knowledge and the empirical statistical models. Mathematical models can benefit the expert systems technology in many ways. When mathematical models are used, the expert systems' decision is constructed based on the physical laws. Although the mathematical models are not exact, their use in the expert systems enhances the performance.

In addition to statistical models, mathematical models can improve the expert systems by declaring what will happen in the future. The speculation of such an event carries some uncertainty. The use of the models within the expert systems is illustrated in Figure 2-21. Steps for use of a mathematical model in the expert systems can briefly be outlined as:

- User enters all the known attributes for the mathematical variables.
- Expert system consults the knowledge base.
- The uncertainties are calculated by the data base statistics.
- The inference engine predicts the future event by using the knowledge base rules, the data base statistics and the result of a mathematical model simulation.

The finding from the knowledge base and the data base are tested by the mathematical model as follows: the known attributes, which the user has entered, are loaded into the simulation model and the output is requested. After calculation of the series of equations, the model simulates the future and calculates an output value. If the model's finding is similar to the

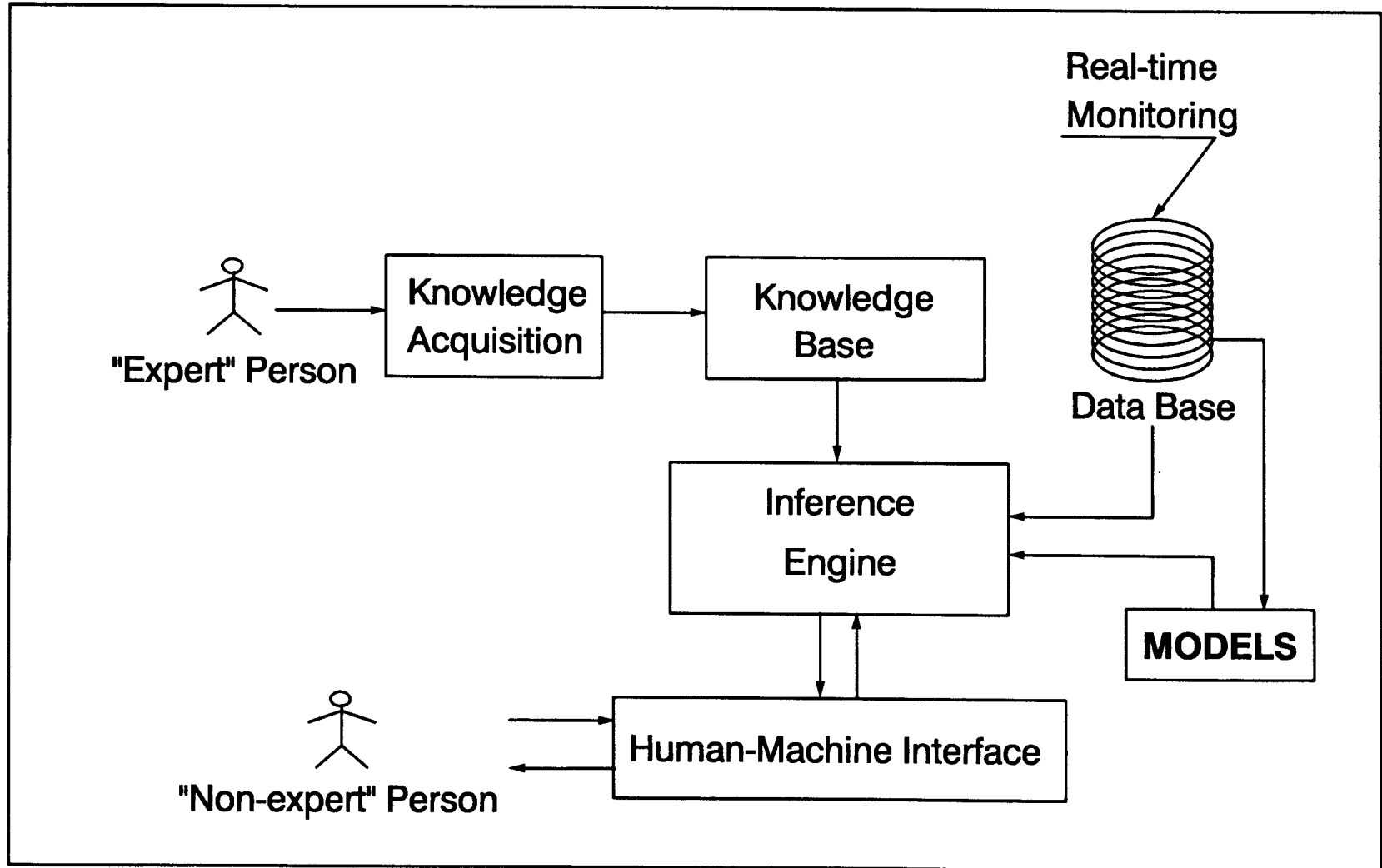


Figure 2-21: Use of mathematical models in expert systems

inference engine's finding, then the confidence factor of that finding increases. In contrast, if the model finds conflicting output, then the expert system decreases the confidence factor for that output.

In some cases, the model can benefit the expert system by identifying the potential erroneous findings. For instance, the inference engine comes up with a solution, but the mathematical model predicts something different from the inference engine's solution, then the expert system is advised to consider the integrity of the solution closely. The expert system goes through the possibilities and picks out the one which all the components of the expert system (*i.e.*, the knowledge base, the data base and the mathematical model components) would agree on. If no such state exists, the expert system adjusts the confidence factors and advises several outputs to the user along with explanations. Use of the mathematical models, therefore, can decrease the possibility of failing within the expert systems.

There are two more common ways to use the mathematical models within the expert systems. First, the mathematical models may serve as a testing tool for the knowledge base. The knowledge which is collected in the knowledge base can be tested by the expert system developer through mathematical models. The developer enters several possible events and compares the result from the knowledge base to the mathematical model's answer. This way, if there is any contradiction, the expert person is consulted and the rules in the knowledge base are modified.

Second, the mathematical models are used to generate rules for the knowledge bases (Olsson *et al.*, 1985; Nolasco *et al.*, 1989). This technique is commonly used if there is an inadequate source of knowledge available for the expert system development. Most expert system developers avoid the interview session during the knowledge acquisition procedure and create artificial rules based on the mathematical model's output. The procedure in practice is as follows: the developer enters fictional events and receives output from the mathematical model. This information is formulated in a rule form and is then added to the knowledge base. After several runs with the mathematical model, the knowledge base is constructed.

Even though the rules are created based only on the mathematical model's efficiency, many applications indicate successful capture of the knowledge by the above mentioned method. For example, in cases when the knowledge engineer misses necessary questions or the expert person does not have reliable knowledge of that specific area, the knowledge extracted from the interview sessions is questionable. This type of doubtful information may be overcome by the help of the mathematical models. If the model is run for the specific condition, valuable information may be obtained.

The mathematical models in the wastewater treatment field are discussed in section 2.1.5. This dissertation will use one of the models for the activated sludge process to enhance the expert system consultation.

2.2.5 Expert Systems Applications

Since the AI researchers have focused on the development of the expert systems, massive work has enabled the expert systems to spread to many field applications. The expert systems can be applied efficiently to any field which requires "expertise." Fields such as medicine, engineering, and education benefit from the advantages which the expert systems offer. In this section, several real-time applications of the expert systems are presented and the future of the expert system applications is discussed.

Several sample categories where the AI has been applied can be briefly shown as: (1) logical reasoning--manipulation of knowledge; (2) expertise--simulate expert person; (3) language--understanding the meaning of human voice; (4) programming--writing computer programs; (5) problem solving--making decisions; (6) learning--teaching to computers; and, (7) robotics--simulate mechanics of human beings (Barr and Feigenbaum, 1981). All the above categories have received comprehensive attention by the AI researchers. This dissertation divides the research field into two categories, the water related fields and the others. First, the water related expert systems are discussed.

2.2.5.1 Expert Systems in Water Quality Management

Early expert systems for water quality were developed in the late-1970s. Since then, the subject has attracted many AI researchers and water quality engineers. The expert systems approach demonstrated successful

applications of improving process control techniques in real-time operations (Beaverstock *et al.*, 1985; Herrod and Tietz, 1989). Table 2-19 lists some of the several expert systems in water and the wastewater fields.

Although all the expert systems in Table 2-19 are worthy of discussion, the following two studies are selected as representative. The *Diagnosis of the Activated Sludge Process* (DASP) (Patry and Chapman, 1989) and the *Statistical and Expert System for Plant Operation* (StatEx) (Berthouex *et al.*, 1989) are chosen because of their similarities to the expert system which is developed in this dissertation. These two studies are investigated separately, in the remainder of this section.

There are two major parts in the DASP. The first part is a rule-based system, *Preliminary Design Expert System* (PDES) (Gall and Patry, 1988), which was created for designing a wastewater treatment plant. The second part is also a rule-based system but it is used to diagnose problems with the activated sludge process and to advice corrective control actions. The PDES's knowledge base helps engineers make the optimum decision on what process to choose for a specific objective within the wastewater treatment. The ability to compare several options provides a great advantage for the design engineers to compare and to choose the right unit processes.

The diagnosis part of the DASP is developed by the knowledge acquisition technique which combines both the rules collected from the literature and the rules which are extracted from interviews with the experienced plant

Table 2-19: Expert systems in water related field

Expert Systems: Description	Usage	References
Wastewater DASP: Diagnosis of problems with the activated sludge process	Diagnosis, Design, Process control	Patry and Chapman, 1989
Diagnosis of wastewater treatment processes	Diagnosis	Johnston, 1985
Wastewater treatment plant operation and control	Process control	Beck <i>et al.</i> , 1978
Diagnosis of an activated sludge plant	Diagnosis	Jenkins and Jowitt, 1987
StatEx: Expert system with data base structure	Diagnosis/Control	Berthouex <i>et al.</i> , 1989
Diagnosis of activated sludge	Diagnosis	Maeda, 1985
Anaerobic digester diagnosis	Diagnosis, Process control	Barnett and Andrews, 1987; Barnett and Andrews, 1990
Pump-Pro	Diagnosis	Fritsch, 1986
Expert system based on deep knowledge	Monitor/Control	Maeda <i>et al.</i> , 1990
Activated sludge operations	Process control	Koskinen, 1989; Koskinen and Viitasaari, 1990
Advanced treatment: Phosphate removal	Process control	Krichten <i>et al.</i> , 1990
Advanced treatment operations	Process control	Kayser, 1990
Activated sludge control using STOUR	Process control	Vitasovic and Andrews, 1987
IOS: Intelligent Operation Support system	Process control	Baba <i>et al.</i> , 1990
Sludge handling	Process control	Nakazato and Okuna, 1990
Wastewater pumping	Process control	Awano <i>et al.</i> , 1990
Activated sludge bulking	Diagnosis	Geselbracht <i>et al.</i> , 1988
SEPIC: Advice on NPDES	Quality control	Hadden and Hadden, 1985
Water Resources Expert advisor for QUAL2E	Parameter selection	Barnwell <i>et al.</i> , 1986
CORMIX1: Advisory program to choose appropriate numerical model	Quality control	Doneker and Jirka, 1988
Reservoir system operations	Process control	Houck, 1985
Updating small water supplies	Design	Knight, 1987
HYDRO: Calibration of watersheds	Quality control	Reboh <i>et al.</i> , 1982

operators. The knowledge base is then revised for the chosen shell, namely the Personal Consultant™ Plus (Texas Instruments, Inc., 1988). The knowledge base consists of 169 rules and 97 attributes which describe the system parameters. The rules are arranged in a manner so that the expert system shell has the ability to disregard unrelated rules at the beginning of the consultation. As a result, consultation time is improved and expert system performance is enhanced. A basic consultation session consists of the following steps:

- User defines a desired knowledge base to use. For example, diagnosis or assessment.
- User specifies requested task, such as consultation or development.
- User enters the type of method--such as observation, measurement, or microscope analysis--which is used to identify the problems.
- User specifies the location of observed problems within the plant. For example, aeration basins, sedimentation tanks, or influent.
- User specifies the problem. For instance, odor, NPDES permit violation, or foaming.
- Expert system starts working.
- Knowledge base is accessed.
- Inference engine is triggered.
- Inference engine pursues the user defined goal. The rules are analyzed and necessary questions asked user to evaluate the rules, for example "Which of the following best describes the appearance of the floc?" and possible answers are outlined as "Regularly Shaped," "Irregular," "Ragged," or "Small" (Patry and Chapman, 1989).

- After the rules are analyzed and questions are answered, the inference engine comes up with an advice or a problem identification.

The original knowledge base was developed for the treatment plant in Oakville, Ontario. The treatment plant consists of grit removal, primary sedimentation tanks, aeration basins, final sedimentation tanks, and chlorination chambers. The real-time application of the DASP was tested on this plant. The authors were not able to make any kind of real-time verification of the expert system because the plant operators did not use the expert system regularly enough to establish a solid background for a verification assessment. Many operators were unwilling to use the computer because they had insufficient experience.

The StatEx, the second sample expert system, is developed by combining a knowledge base and a data base component within the expert system's structure (Berthouex *et al.*, 1989). The knowledge base contains 13 rules which mainly deal with the activated sludge process. After the data base is constructed by collecting real-time application data, a statistical model is applied to the data base variables to obtain helpful information for expert system components. The program operates as follows:

- User enters the necessary data. There is a special element within the program which gives the user ability to enter data in a menu-driven fashion.
- Given data is analyzed. Consistency, missing parts, and integrity of data is checked.
- Advisory expert system session starts.

- Knowledge base is accessed.
- The fuzzy terms are identified by the statistical analysis of related historical data. The words like "high" and "low" are given a mathematical value by analyzing frequency of that specific attribute.
- Inference engine updates knowledge base whenever new data is given to represent the most recent characteristics.
- Data are validated by control charts.
- Finally, the inference engine advises the optimum process control strategy at the end of consultation.

The development of expert systems along with a data base is a promising method. Future research, undoubtedly, will expand the use of real-time values within the expert systems.

2.2.5.2 Expert Systems in Other Fields

Expert systems have been extensively used in other fields. Levitt and Kartam (1990) discussed expert system use in the construction industry. Many examples of expert systems applications in the control system engineering are given in Linkens (1990). Additional fields of expert system applications include education, environmental engineering, and geological engineering. Table 2-20 shows some expert systems which have been developed for these fields.

One expert system from Table 2-20 will briefly be discussed here: MYCIN (Shortliffe, 1976). MYCIN has given many AI researchers hope about the

Table 2-20: Expert systems in other fields

Expert Systems:Description	Usage	References
<u>Medicine</u>		
MYCIN: Diagnosis of bacterial disease	Diagnosis	Shortliffe, 1976
<u>Science & Industry</u>		
GEOTOX	Hazard Assessment	Mikroudis <i>et al.</i> , 1986
RPI: Site assessment system	Assessment	Law <i>et al.</i> , 1986
FALCON: System failure diagnosis	Diagnosis	Rowan, 1987
DEMOTOX	Hazard Assessment	Ludvigsen <i>et al.</i> , 1986
ESCORT	Real-time operations	Sacks <i>et al.</i> , 1986
PSI: Automatic programming	Programming	Palmer, 1985
REVA: Diagnosis of problems with rotating equipment	Diagnosis	Finn and Hall, 1986
FRES: Chemical spill clean-up advisor	Hazard Assessment	Hushon, 1986
HOWSAFE	Evaluation of safety	Levitt, 1986
Waste management advisor	Hazard Assessment	Patry and Chapman, 1989
PROSPECTOR: Mining advice	Pattern recognition	Duda <i>et al.</i> , 1978
EXACT: Self-tuning control via pattern recognition	Process control	Dearden, 1989
Incinerator Diagnosis	Assessment	Huang <i>et al.</i> , 1986
Incineration systems	Process control	Yang, 1989
Geophysics system	Analysis	Patry and Chapman, 1989
PROP: Evaluating chemical contamination in power plants	Process control/ Diagnosis	Gallant <i>et al.</i> , 1985
FLEX: Liner advisor for hazardous waste material	Diagnosis	Rossmann and Siller, 1987
<u>Education</u>		
WHO	Teaching	Palmer, 1985
SCHOLAR	Teaching	Palmer, 1985
<u>Legal</u>		
DISCON: Differing Site Conditions Clause	Legal Advice	Kraiem <i>et al.</i> , 1989

future of expert systems. The MYCIN's knowledge base is designed to identify bacterial diseases. The knowledge base is made out of rules which contain pre-fixed confidence factors. Typical consultation steps are as follows:

- Physician enters findings of tests on the isolated microorganism from a patient. For example, findings could be gram positive, or the shape of the organism.
- Expert system starts working.
- Knowledge base is analyzed under given attributes from consulting physician.
- If more information is needed to make a decision, the inference engine asks questions of the physician to specify findings.
- The inference engine reaches a decision and the physician is advised of the possible causative bacteria by declaring confidence factors to each advice.

Much interest has been shown in MYCIN because its performance is comparable to that of the most experienced physicians. Uncertainty calculations in MYCIN have become a landmark for other developers to test their uncertainty models. Some researchers have demonstrated the significance of the way the probabilities are calculated by comparing to the MYCIN-like models (Fox *et al.*, 1980; Neopolitan, 1986). Their finding indicates the importance of using different techniques to calculate the confidence factors. This dissertation uses the method which is explained in section 2.2.2.7.

2.2.6 Selected Shell: Personal Consultant™ Plus

After an extensive research of expert system shells, the Personal Consultant™ Plus shell, Version 4.0 (Texas Instruments, Inc., 1988) for the DOS environment was chosen to be the most suitable expert system shell for developing a knowledge-based expert system for the control of a selected refinery activated sludge plant.

The Personal Consultant™ Plus (PC Plus) shell is fast and capable of accomplishing many difficult tasks. The PC Plus shell was selected for the following reasons:

- *Capable of handling multiple-frame knowledge bases:* For complex knowledge bases, it is necessary to separate the rules under individual sub-frames. The multiple-frame organization provides a more understandable, and serviceable knowledge base. Furthermore, the use of multiple sub-frames significantly improves the organization of the knowledge and the overall quality of the knowledge base.
- *Ability to manipulate sub-frames individually:* Not all the sub-frames need to be accessed during an expert systems consultation. Avoiding certain sub-frames and manipulating the selected sub-frames individually improves the expert systems speed and eliminates the unnecessary questions asked during the consultation session.
- *Manageability of the order in which the rules will be evaluated:* The UTILITY function allows the developer to define the sequence of the rule testing process. If there is more than one rule which concludes the same parameter, the UTILITY number controls the order in which the rules will be tested. Placing the correct UTILITY number significantly improves the computational speed.

- *Ability to use external DOS programs:* Although, both the mathematical and the statistical models are developed separately from the knowledge base, the expert systems accesses these models during the consultation process. The input from these models are the backbone of the knowledge base accuracy. The models are written and executed separately through the DOS environment. The use of these models is essential.
- *Capability of communicating through text files:* Both the models and the user-interface communicates with the ASCII text files. Input and output parameters are stored in text files and used by the expert system shell. The use of text files yields a friendlier user-interface and an improved overall performance of the expert systems.
- *Ability to generate an independent running copy:* Independent running copy of the knowledge base is faster because it is separate from slow PC Plus development environment. Additionally, the independent running copy lowers the software cost because it eliminates the need for purchasing the PC Plus development package, separately. Furthermore, the running copy of a knowledge base occupies limited space in a hard-disk. The small size provides space for the rest of the expert system programs.

In order to write the rules with the PC Plus shell in this dissertation, the logic trees are created using the nomenclature shown in Figure 2-22. The figure illustrates the control of pH which specifically tests the pH and the amount of pH controlling agent addition. The objects are designated by an elliptic shape, the attributes and values are shown on the arrow which links two objects.

Each rule has an assigned number for identification and segregation purposes. The rules are entered and modified under the PC Plus development environment. The PC Plus' internal text editor and error testing compiler is capable of generating complex knowledge bases.

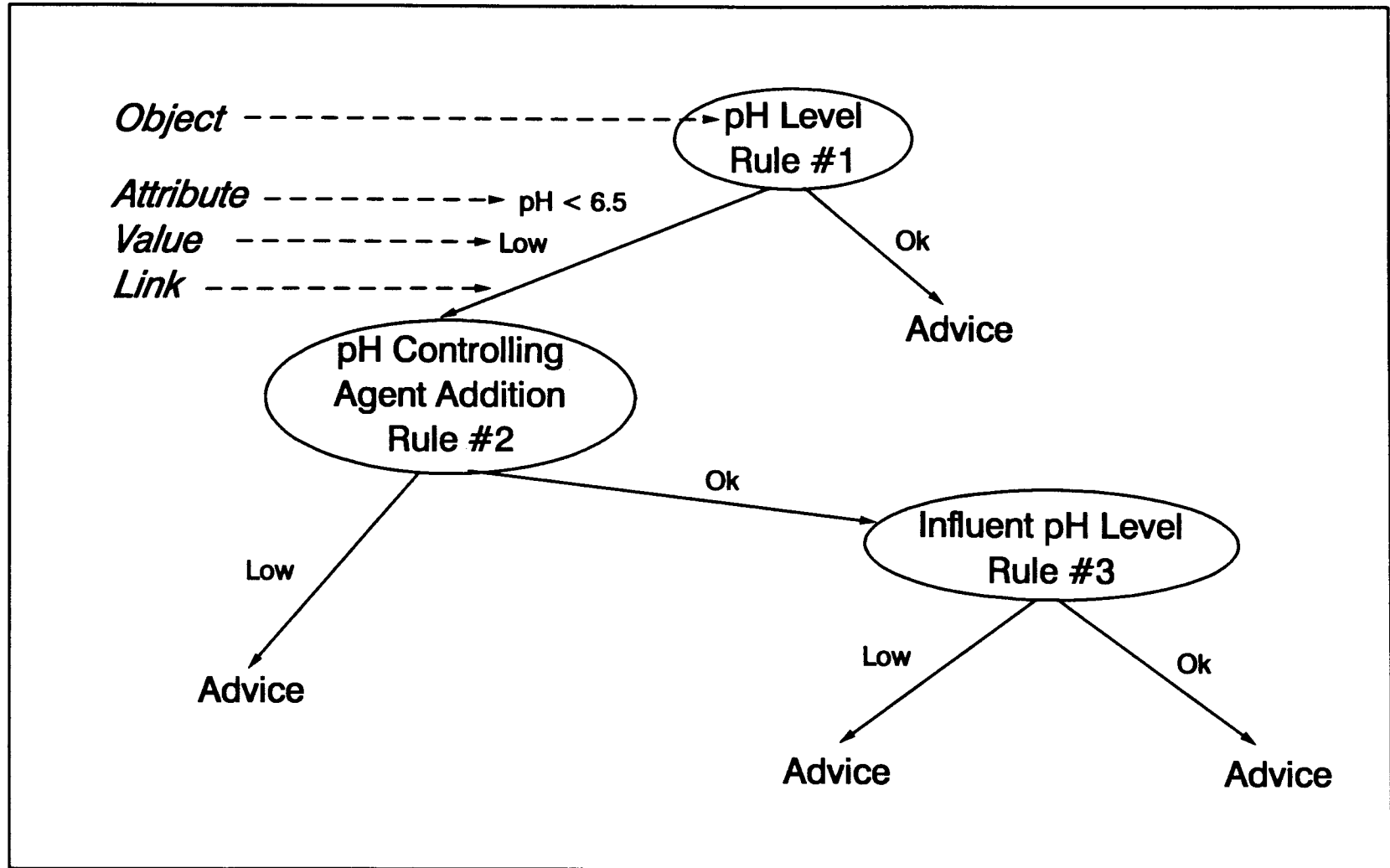


Figure 2-22: Example of nomenclature used in the logic tree creation

The major functions of the PC Plus' inference engine are outlined as follows: (1) identify parameters which will be searched; (2) put parameters in an order to provide the most efficient search procedure possible; (3) designate the rules which will be used during the search process. The default search algorithm of the PC Plus' inference engine is the backward chaining (see section 2.2.2.6).

The remainder of this dissertation will use the PC Plus expert system shell to develop a knowledge base for the process control of nitrification in the selected refinery process-water activated sludge treatment plant.

3 Development of the Activated Sludge Process eXpert (ASPX)

In the remaining portion of this dissertation, *Activated Sludge Process eXpert* (ASPX) will be developed. ASPX is a knowledge-based expert system designed exclusively for the control of the selected refinery activated sludge plant. The primary task in developing the ASPX is as follows: to create an expert system that will perform equal to an expert plant operator who has 20 years of practical experience in the selected plant.

Achieving this task is challenging since development of expert systems involves highly complex and intricate preparation procedures. A successful implementation of computer programs requires the following steps: (1) long-range hardware and software planning; (2) identifying the system requirements; (3) software design and development; (4) software testing; and (5) installation and training (Laws, 1991). The development of the ASPX will be achieved through the following steps:

- 1) Building the foundation of ASPX which consists of:
 - a) Knowledge acquisition process.
 - b) Writing the knowledge base.
 - c) Data base conversion.
 - d) Development of statistical and mathematical models.
 - e) Formation of the user-interface program.

- 2) Testing ASPX which involves the following internal steps:
 - a) Initial testing.
 - b) Implementation of ASPX to the refinery plant.
 - c) Intermediate testing.
 - d) Advanced testing.

3) Process control of the refinery wastewater treatment plant by ASPX.

In the first step, the core of ASPX is created. The knowledge collection, coding, and organization is performed. The data base and both the statistical and mathematical models are established to obtain real-time variables for the expert system. The creation of the user-interface gives inexperienced users ability to access the full range of ASPX program capabilities.

The second step involves testing and implementation of ASPX into the selected refinery plant. The testing is performed by both the developers and the operators in the plant. The 3-step testing procedure is used to improve the accuracy of the ASPX program and to enhance the user-interface portion. The first step in the 3-step testing is verification of the rules in the knowledge base by the developer. The rule structure, organization, and knowledge gaps are identified and corrected. The second step in the 3-step testing consists of validation of the rules. The procedure is performed by an experienced plant operator by entering hypothetical conditions and evaluating future predictions. In this step, the operator can identify rules that lead to incorrect suggestions. The third and final step in testing is performed by applying ASPX to the real-time operations. The final advanced testing will reveal last modifications required for the rules and the models.

The implementation procedure, in the second step, consists of setting up the computer in the plant and giving a brief course to the plant operators on the use of the ASPX program. The user-interface is further modified to satisfy the demands of the operators.

The third and final step is to put ASPX in charge of the refinery plant operations. Under supervision, ASPX operates the plant and provides a powerful tool to the operators to control the plant operations. The real-time application of the program will reveal the full capability of the ASPX program.

The sections of this dissertation focus on the steps in developing the ASPX program. Subjects covered are: knowledge acquisition technique, knowledge base structure and construction, handling uncertainty, data base usage, ASPX programming, testing procedures, uses of ASPX, and finally application of ASPX to the selected refinery activated sludge process.

3.1 Building the Foundation of ASPX

The initial setup of the ASPX program involves considerable time and effort. Building the foundation of ASPX is achieved by the following five steps.

First, identify the type and source of knowledge required. The knowledge sources include the literature, university faculty members, refinery plant operators, and the author's individual knowledge in the domain.

Second, obtain the required knowledge, and create the logic trees. The logic trees represent how the knowledge is linked together. The integrity of the knowledge base lies in the development of the logic trees. The pieces of knowledge in the logic trees are arranged in a manner allowing for an uninterrupted flow of knowledge upon completion of the tree.

Third, code the logic trees into the knowledge base in a rule form. The knowledge is coded based on its location in the logic tree. The coded knowledge is entered into the knowledge base in separate categories. Parameters to be used in the expert system are identified and entered into the knowledge base. The confidence factors are assigned to the rules based on the degree of reliability.

Forth, set up the data base functions, including the models. The historical data base is converted to the new format which enables more efficient use of the process data. Statistical and mathematical models are developed to analyze the historical data and to predict future events. The models provide an invaluable tool for supervising the expert system's advice. In case of erroneous advice given by the expert system detection will be made by the models.

The fifth and final step in building the foundation is to generate an initial user-interface program. The user-interface is designed for operators who are less familiar with computers. The user-interface enables operators to utilize the full functions of ASPX without comprehensive computer experience.

Each of the steps required to construct the foundation of ASPX is documented in the following sections.

3.1.1 Knowledge Acquisition Technique for ASPX

Primary knowledge is required about the nitrification process. The knowledge should include the diagnostic information on abnormalities which are causing nitrification failure, and the corrective information which gives remedies for the failure.

The required knowledge is first compiled from the literature by the author, and then modified by a university faculty member who has a considerable standing in the wastewater treatment field. Both the diagnostic and corrective knowledge are combined and categorized for the development of the knowledge base.

The categorization of the knowledge consists of the major parameters which affect the nitrification process. The parameters chosen are as follows: dissolved oxygen, pH, sludge age, temperature, ammonia, and

phosphate concentration. Each parameter is arranged under a separate sub-frame (see Figure 3-1). The pieces of knowledge are distributed to the sub-frames based on corresponding categories. Within the sub-frames, the knowledge is further arranged and the logic trees are generated. For each sub-frame, there is one logic tree which links the pieces of knowledge together on the sub-frame's category. For example, the category pH in the Figure 3-1 has one logic tree which contains the pH parameter, influent pH, and the addition of caustic solution.

The formation of logic trees is a critical stage in developing an expert system because the logic should neither have any gaps nor contain any incorrect knowledge. The integrity of the whole expert system depends on the logic tree setup. The generated logic trees are carefully analyzed by both the author and the university faculty members. After many modifications, the final format of the logic trees was accepted. The full representation of the logic trees which are used in this dissertation may be found in Appendix A and the explanation of the result codes are shown in Appendix B.

Further modification to the rules may have been performed by the result of the interviews with the experienced operators. However, in this dissertation, the interview sessions are omitted because of the difficulty in finding experienced refinery plant operators, and the belief that the developed logic trees are capable of accomplishing the required tasks from the ASPX.

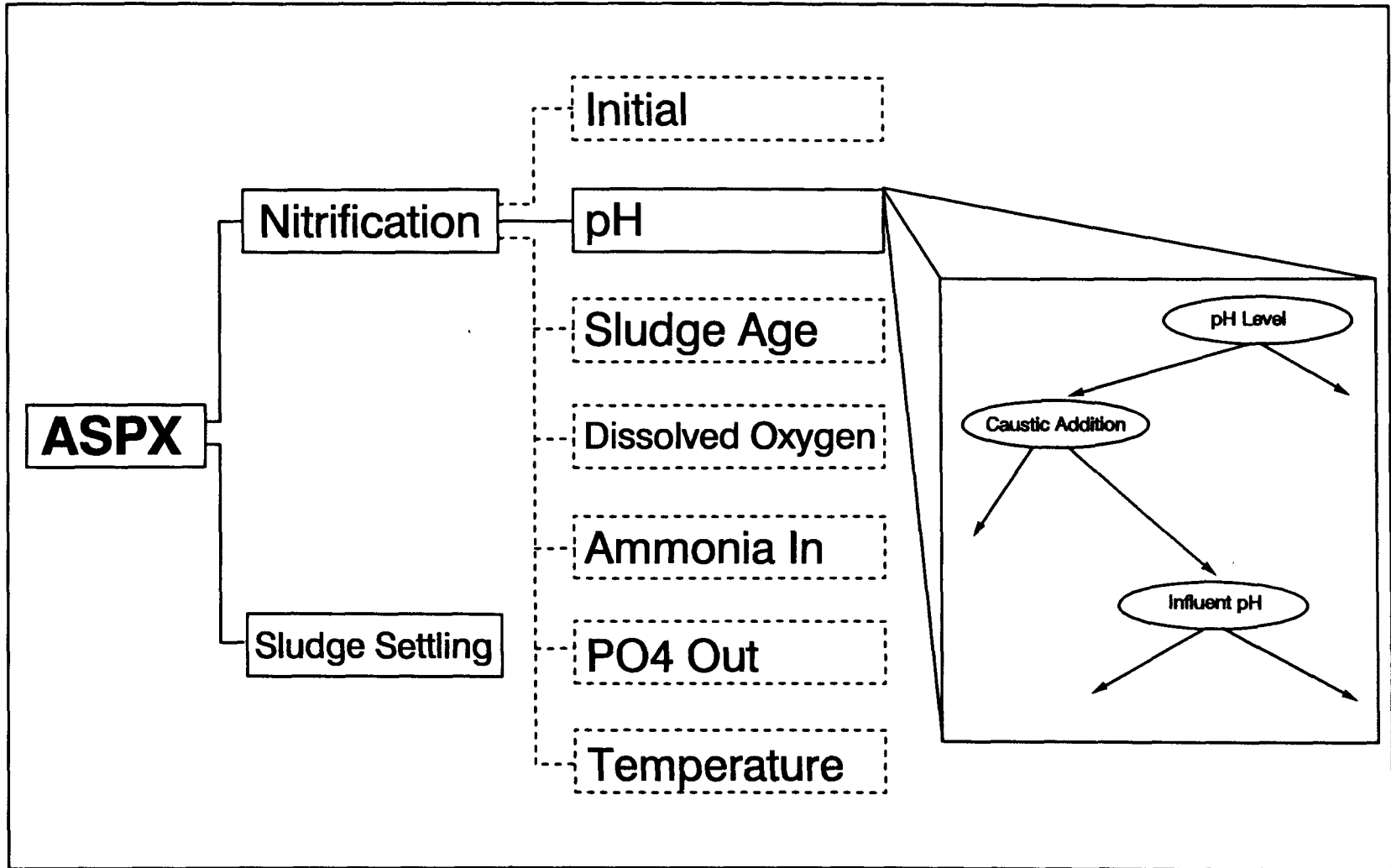


Figure 3-1: The frame structure of the ASPX knowledge base

3.1.2 Structure of ASPX

ASPX is a knowledge-based expert system which contains all the parts of a typical expert system program. The structure of ASPX consists of the following parts: expert system shell--PC Plus; knowledge base; data base; statistical, and mathematical models; innovative user-friendly interface; and report generation module (see Figure 3-2).

The flowchart of ASPX is shown in Figure 3-3. The operator accesses the data base module through the user-interface. The mathematical and the statistical models read input parameters from the data base, along with the parameter reading module which transfers the required data to the input files. The input files consist of both the parameters and the results of the model simulations.

The data from the input files are used by the rules in the knowledge base. The PC Plus shell accesses the knowledge base and based on the input variables, makes a decision. The results are stored as codes in a separate text file. The results are then decoded and fed into the user-interface where the operator can review the results. The coding/decoding is necessary to eliminate large-size knowledge bases and to allow possible modification of the results by the user.

Because modification of the rules in the PC Plus shell requires considerable experience with computers, the input parameters used in the knowledge base are kept "flexible." The operator is able to change the input variables within the rules through the user-interface without altering

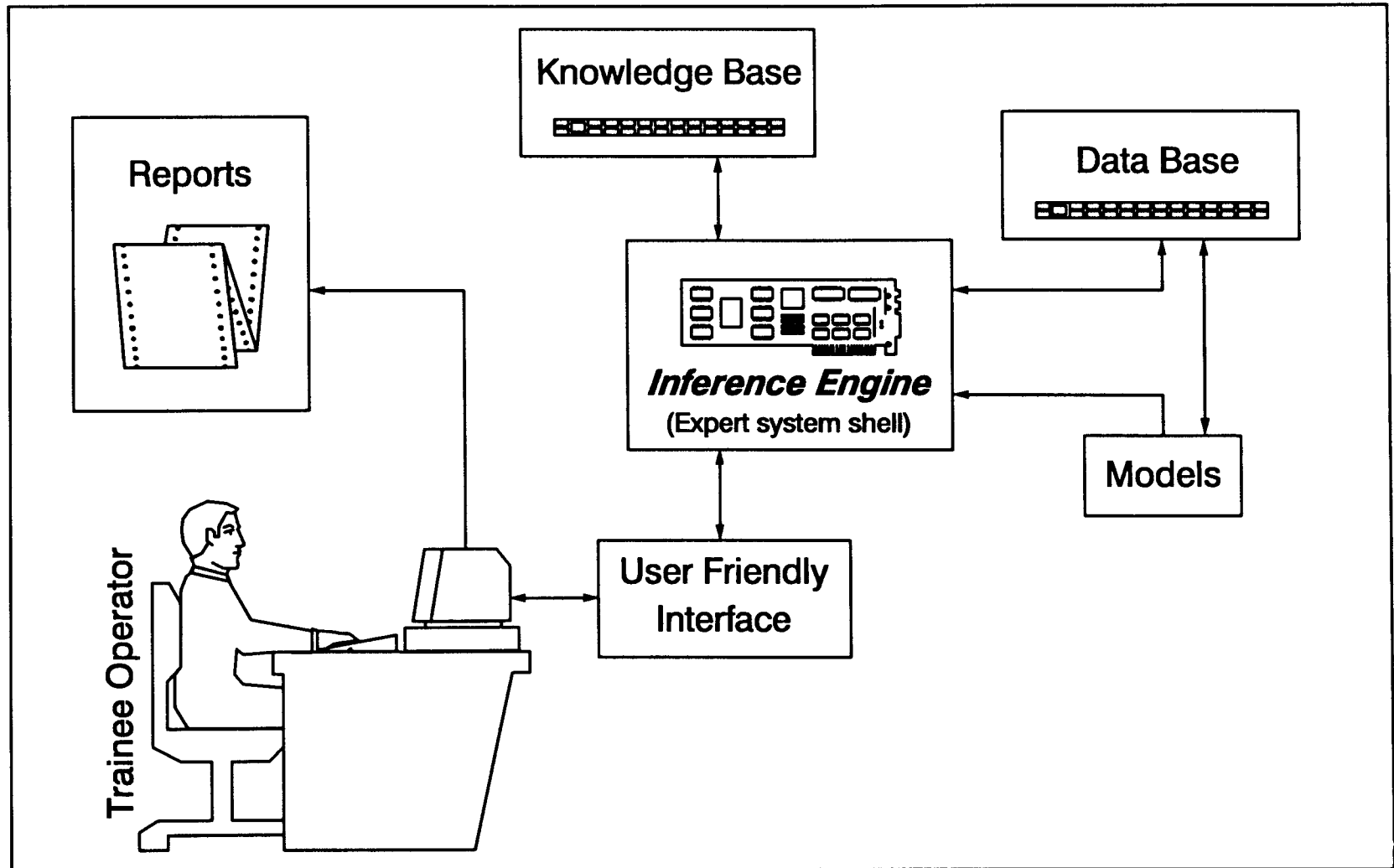


Figure 3-2: Interaction between the different modules of the ASPX program

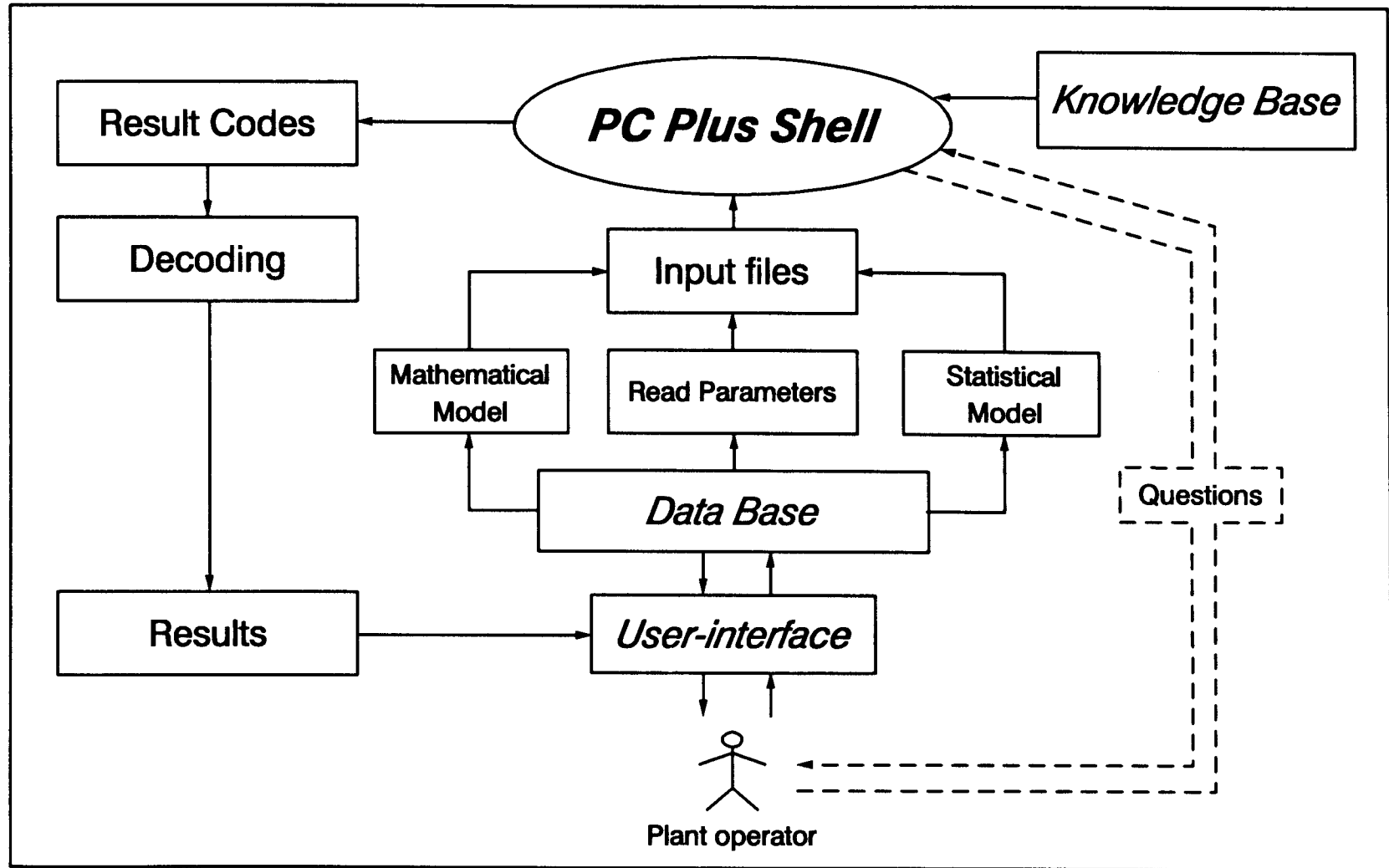


Figure 3-3: The flowchart of the ASPX consultation

the original knowledge base rules. The values for the parameters are stored in a text file where the PC Plus reads these values at the beginning of each consultation session. This flexibility of the input parameters eliminates the need for learning the PC Plus development program.

3.1.3 ASPX Knowledge Base

After the knowledge acquisition process, and the determination of how the components of ASPX will be installed, the knowledge base is constructed. The knowledge base is built on the root-frame.

The organization of the sub-frame determines how the parameters and the rules will be used. ASPX's frame structure is as follows (see Figure 3-4): the root-frame is named ASPX. The sub-frame "Nitrification" and "Sludge Settling" are child frames of the ASPX root-frame. The "Sludge Settling" sub-frame has a limited number of rules and is only intended to be used as an indicator for sludge settling problems. There are seven sub-frames of the Nitrification sub-frame. The first sub-frame of the Nitrification sub-frame, "Initial," has rules to transfer input parameters from the input text files. The remaining six sub-frames represent the parameters which significantly influence the nitrification process.

Each sub-frame is stored in separate files in a hard-disk. The files are indicated with extension "KBName.K#". KBName is the name of the knowledge base, and the sign "#" indicates the number of the sub-frame.

		Rules	Suggestions
ASPX	Nitrification		
	Initial	5	0
	Ammonia	44	22
	Sludge Age	104	54
	Dissolved Oxygen	100	51
	pH	26	13
	PO4 Out	26	13
	Temperature	24	12
Sludge Settling		8	8
Total:		337	173

Figure 3-4: ASPX sub-frames and the number of rules and suggestions for each sub-frame

For example, the temperature sub-frame is stored under name "ASPX.K8", and the pH sub-frame is stored under "ASPX.K2".

Each sub-frame has the following properties: identifier, parents, parameter group, rule group, sub-frame parameters, and sub-frame rules. The identifier differentiates the sub-frames. For example, the pH sub-frame is identified as "PH-". The parents are declared for each sub-frame because the inheritance plays an important role during the rule evaluation process. For example, the pH sub-frame may only be accessed through either the ASPX root-frame or the nitrification sub-frame (*i.e.*, pH sub-frame's parents).

The parameters are categorized under several groups. Each sub-frame has one parameter group which includes all the variables used in the individual frame. For example, the pH sub-frame has a parameter group "PH-PARMS" which contains parameters about the pH. Similarly, the rules are compiled under categorized groups for each sub-frame. For example, the pH sub-frame has the following setup:

```
Frame _____ :: PH
IDENTIFIER      :: "PH-"
PARENTS         :: (NITRIFICATION ASPX)
PARMGROUP      :: PH-PARMS
RULEGROUPS     :: (PH-RULES)
PH-PARMS       :: (PH-LEVEL CAUSTIC INFLUENT-PH)
PH-RULES       :: (RULE004 RULE005 RULE006 RULE007
                  :: RULE008 RULE010 RULE013 RULE017)
```

The parameter is a variable which is used within the rules. A parameter has three major properties: TYPE, UPDATED-BY, and USED-BY. The TYPE property of a parameter declares what kind of parameter it is. The TYPE of parameters are: SINGLEVALUED (only one value), YES/NO (value is either yes or no), MULTIVALUED (more than one value), and ASK-ALL (all given values are asked of the user).

The second major property of a parameter is UPDATED-BY which contains the rule numbers that conclude the parameter. The list of the rules, which contain the parameter in its *THEN* (...) block, is compiled under the UPDATED-BY property. When the parameter is searched, the inference engine evaluates the rules in the UPDATED-BY list.

The third major property of a parameter is the USED-BY component. The rules which contain the parameters in its *IF* (...) block are identified and cataloged under the USED-BY module. The USED-BY property enables the developer to review the list of the rules which are involved. The identification of the rules which are involved significantly improves the development of the knowledge base in regards to both accuracy and speed. A sample parameter is shown below with its three major properties:

PH-PARMS

```
PARAMETER  :: PH-LEVEL
TYPE       :: SINGLEVALUED
UPDATED-BY :: (RULE005 RULE007 RULE009 RULE011)
USED-BY    :: (RULE004 RULE006 RULE008 RULE010)
```

The values may be assigned to the parameters in several ways. First, the value is read from the input files and assigned to the matching parameters. Second, the value may be assigned by the user directly answering the questions asked by the inference engine. Third, the values are declared by the mathematical operator, "=" . Or, fourth, if a value for the parameter cannot be established, the default value is assigned to the parameter.

In the ASPX knowledge base, the values are assigned through the DOS text files. At the beginning of each consultation, the PC Plus shell reads the input values from the input files, and assigns the corresponding values to the parameters. Flexibility of changing values through the text files eliminates the need of altering the original knowledge base rules. The user has the power to assign any value to each parameter within the knowledge base. Because the user does not alter the rules, the possibility of a system crash is abolished.

3.1.3.1 Rules in the ASPX Knowledge Base

Formulation of the knowledge follows the completion of the knowledge base configuration and the sub-frame setup procedure. The required knowledge for the control of the selected refinery activated sludge plant is formulated into rule forms, and incorporated into the knowledge base structure. See section 2.2.1 for detailed explanation of the rule structure.

The PC Plus shell uses either the LISP or the Abbreviated Rule Language (ARL) form for writing the rules. The ASPX knowledge base consists of

337 rules which are written in ARL form. The number of rules for each sub-frame in the ASPX knowledge base is presented in Figure 3-4. The five rules in the initial sub-frame are used to assign values to the parameters. The rules in other sub-frames use the values transferred by the initial sub-frame to reach a conclusion. A total of 173 different result codes can be generated by the ASPX. A sample rule shown below evaluates the pH level and determines the result code.

RULE005

If the current pH level is less than the minimum allowed pH value,
Then it is definite that the result code is 32.

The rules are organized based on the logic trees developed during the knowledge acquisition process. Within the sub-frames, the rules are arranged in a manner that will provide a fast and accurate information flow. The sequence in which the rules are evaluated is controlled by the rule's UTILITY property. If there is more than one rule which concludes the same parameter, the use of the UTILITY property would provide pruning of the unnecessary rules by placing the most likely rule first. The rule organization is performed by the following steps:

- The knowledge is represented in logic trees during the knowledge acquisition process. The flow of logic indicates how the rules are written and organized. Writing the rules starts from the bottom of the tree and works its way up until the top is reached.
- The rule numbers are assigned in the order which they are created.

- Within the sub-frames, the rules are further arranged by assigning the UTILITY numbers. The UTILITY property controls the sequence in which the rules are fired.
- The result codes are assigned based on their order of appearance within the logic trees.

The forward chaining rules are declared by the rule's ANTECEDENT property. If the ANTECEDENT property is set to "true," the inference engine treats the rule as a forward chaining rule. The forward chaining rules in ASPX are used to display information to the user.

After developing the knowledge base rules, the confidence factors are declared for each rule. The confidence factors within the knowledge base determine the credibility of the rule's conclusion. Handling the uncertainty in the ASPX knowledge base is discussed in the following section.

3.1.3.2 Handling Uncertainty in ASPX

The rules and the parameters in the ASPX knowledge base contain confidence factors to indicate the degree of reliability about the acquired knowledge. The CFs are included in ASPX by the following methods: default CF assignment to the parameters, pre-assignment of the CFs to the rules, CF assignment by the user responds, and data base CF assignment.

The default CFs for the parameters are determined by the accuracy of the measuring equipment for that parameter. The accuracy of the equipment

is influenced by both the laboratory and the equipment conditions. Since most of the process control data for the treatment plant is measured in the field, further reduction of the accuracy is observed. The error made during the laboratory and the field measurements, and the default CFs for the parameters are shown in Table 3-1.

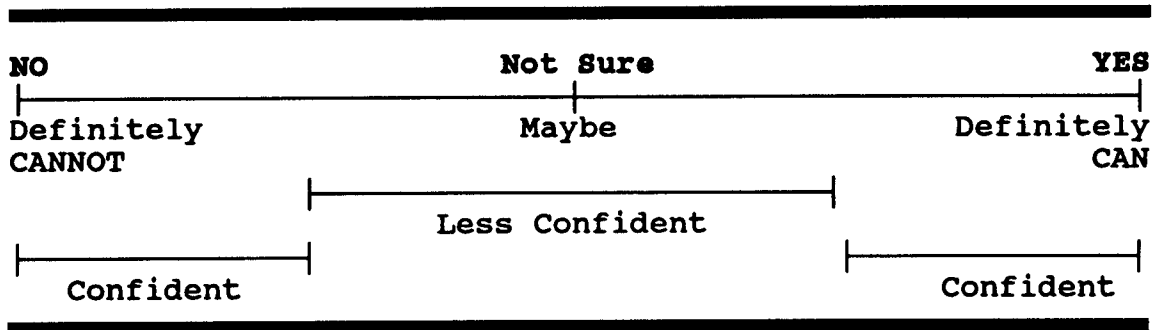
Table 3-1: Error and the default CFs for the input parameters in ASPX

Input parameters for ASPX	Error during measuring process*	Field Adjustment to the Error	Default CF
Flowrate	3%	10%	CF 90
Dissolved oxygen	8%	10%	CF 90
pH	5%	8%	CF 92
Solids (MLSS)	10%	13%	CF 87
Ammonia concentration	10%	13%	CF 87
Phosphate concentration	10%	13%	CF 87
Temperature	3%	5%	CF 95

* Estimated based on the laboratory measurements reported in APHA *et al.*, 1989

Pre-assignment of the CFs to the rules is achieved during the knowledge acquisition process. The pre-assigned CFs reflect the reliability to the knowledge which is formulated into a rule form. The words which are used to describe the knowledge, have pre-determined CF values. Words such as "likely," "maybe," and "sometimes." conceal specific uncertainty

addition, the user enters the number and includes "CF 90" to indicate the 10% uncertainty of the value. If a YES/NO question is asked, such as "Can you divert part of the influent water for at least 2 hours?," the user may indicate the uncertainty by responding based on the following scale:



The confidence factor may also be assigned by the data base statistical model. The historical data represents the behavior of the plant which can be used to calculate the CFs for the rules. In the previously mentioned example, the knowledge stated that when the pH level is below the optimum value, the nitrification often fails. Same criteria may be evaluated by the statistical data base model to improve the accuracy on the CF assignment. When the data base is analyzed, the knowledge is updated as "when the pH level was below the optimum value, the nitrification has failed 23 times out of 25 (92%)." The new CF 92 is assigned to the rule's conclusion so that the accuracy of the expert system advice is improved. Further adjustment is possible by the use of Bayesian statistics which predicts the future CF value. For the same knowledge, the Bayesian statistics updates the CF as "If the pH level drops below the optimum value, the nitrification will fail with a 95% chance." The adjustments on

the CF using the future predictions introduces a new line of accuracy in the expert systems technology.

After the confidence factors are entered into the knowledge base, the PC Plus inference engine manipulates the CFs as described in section 2.2.6.

3.1.4 ASPX Graphical User-Interface

The ASPX graphical user-interface is designed for computer illiterate users to access the ASPX's full options. The ASPX user-interface is a Windows-like, a menu-driven, a mouse/keyboard-controlled, and a high graphical interface program which exposes all the features in ASPX that otherwise would have been inaccessible by an average user. This high-performance ASPX user-interface makes the expert system easy to use for any one in the plant personnel.

The minimum hardware requirement for running ASPX is a DOS-based PC having a 386 processor running at 20 MHz speed or greater with a 13" VGA screen, and a mouse. The hard-disk requirement is only 4 MB for the program and an additional 4 MB for data base files. A large memory (*i.e.*, 4-6 MB) is desired since the PC Plus shell runs faster on the large extended memory configurations.

The ASPX user-interface program is written in Turbo Pascal language version 5.5, using the Video Graphics Array (VGA) resolution graphical

presentation. The interface program require a program code of 3,000 lines. The Turbo Pascal language is chosen because it has advantages over other algorithmic languages such as FORTRAN and BASIC which would only provide limited graphical functions and limited interfacing properties. The Turbo Pascal language is capable of performing complicated simulation model equations fast. Furthermore, the record type in the Turbo Pascal enables numerous advantages in data base record handling including faster access, less memory requirement, smaller file sizes, and more efficient manageability of the records.

The interface program is a graphical menu-driven program where the commands are given by either a keyboard or a mouse. Although using the arrow keys and the enter key on the keyboard is an alternative, the use of a mouse is recommended because it provides a more efficient control of the interface program. The menus indicate possible alternatives for an easy access to different functions of the ASPX program. The user simply points the selection by a mouse and presses the mouse button to make a selection. The mouse controlled menus and icons are a simple way to unveil ASPX's full potential.

The need for learning the PC Plus program is eliminated by the introduction of the user-interface program. Otherwise, the use of the PC Plus shell requires a considerable amount of experience with the program where the frustration may discourage the user from using the expert systems. Therefore, construction of a user-friendly interface for the ASPX

program is an essential step in the development of the expert systems application.

The ASPX's interface program consists of the following three major sections: (1) data base manipulation; (2) expert systems; and (3) report generation (see Figure 3-5). The first part is the data base handling where the user performs a number of data base related functions including adding or changing data base records, pattern analysis, graphical representation of the records, data base statistics, transferring data base files, and confirmation of the data base records.

The second function of ASPX is the expert systems where the logic trees are analyzed based on the plant's current data. The expert systems functions consist of setting low and high values for the parameters, displaying the nitrification state, running the PC Plus shell, and utilization of the results.

The third function of ASPX is generation of reports required by the plant management. The reports include daily activity reports showing the process control parameters and how they are controlled during the day. Additional reports such as the operator activity log, daily turnover sheet, and graphical presentation of data base records may be generated by the ASPX program. The internal clock notifies the operator to generate a report for reporting and record keeping purposes. If the operator has been reminded of and did not generate a report, the supervisor is notified for possible violation of compliance with the plant record keeping practice.

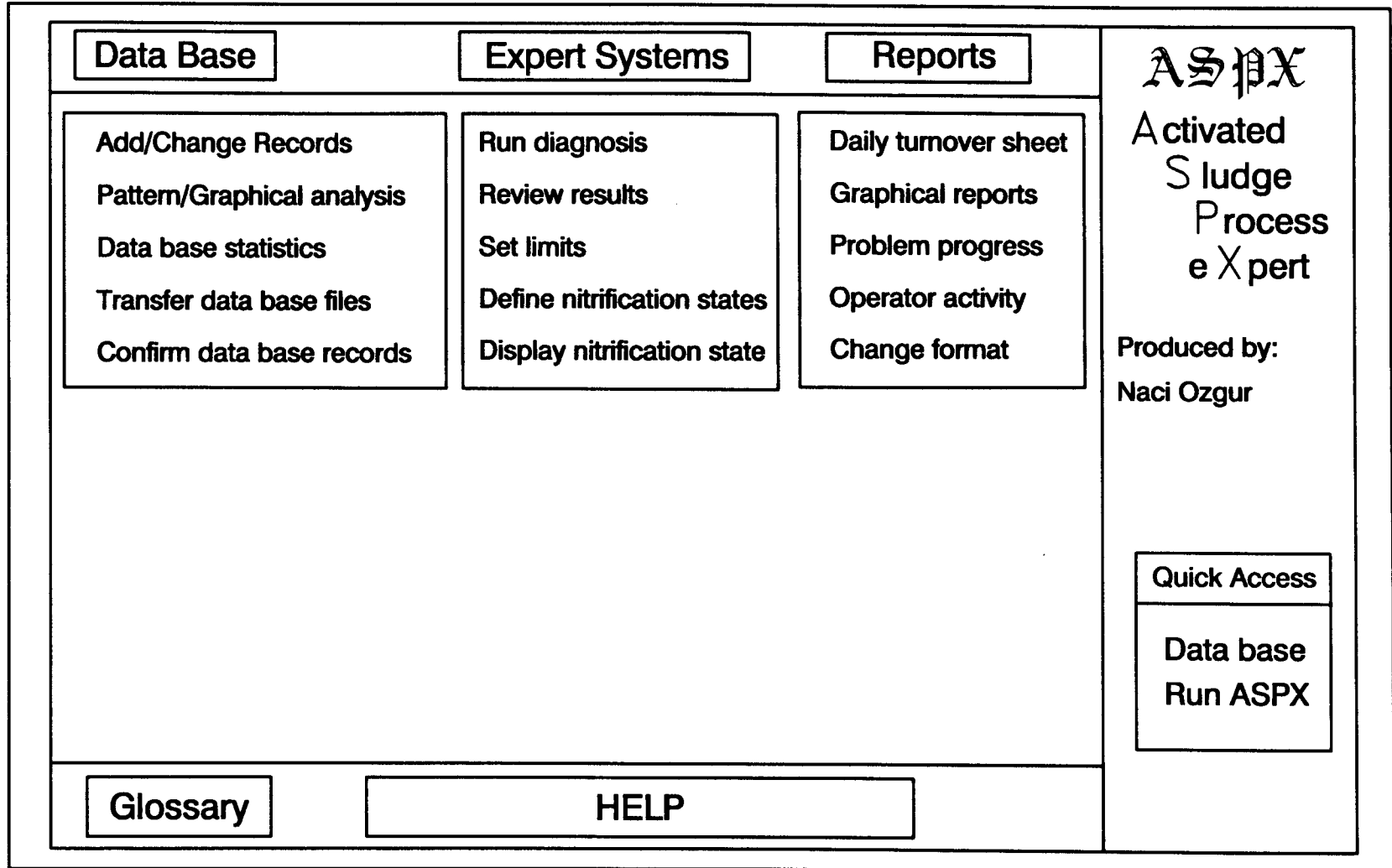


Figure 3-5: A sample screen from the ASPX program showing the three main menu options

Furthermore, since the operator activity log will keep track of what the operator is doing, the report generation by the computer discourages possible criminal offense intentions such as falsifying the records.

For the enhancement of the user-interface, a questionnaire which asks of the selected plant operators' input was distributed. The questionnaire covered the areas such as operations, monitoring, problem diagnose techniques, and data base record handling. The responses obtained from the questionnaire were used to make modifications to the user-interface. The following enhancements were made based on the suggestions given by the operators: (1) include more lab-based measurement results; (2) give site-specific directions to the operators such as measure DO at the head end of the basins, or perform leak detection test using the kit sold in the maintenance shop; (3) dedicated lab technician is needed to perform regular monitoring; (4) include statistical values such as typical value, value considered abnormal, and last week's pattern; (5) determine the frequency of problems the plant experiences; and (6) what the operators expect from the computer and whether they are willing to use it. The complete questionnaire responses are provided in Appendix C.

3.1.5 ASPX Data Base

The data base component of ASPX has a significant function within the program to provide a link between the theoretical knowledge and the real-time plant data. General theoretical knowledge alone may not be

representative of the selected treatment plant. The data base module provides a customized process setup for ASPX at the selected refinery treatment plant. Prior to the application of ASPX, the old data base records are converted to the new format which allows faster access to the data, and smaller size of data base files.

The selected refinery plant has been keeping data base records since the beginning for their operations. Currently, twenty-one parameters are being measured with both the field and the laboratory equipment, 3 times a day with 8 hour intervals. Table 3-2 shows the individual properties of these parameters including the type, the location, and whether it is a required or an optional parameter. While the type of measurement effects the default CF values, the location of the measurement indicates the presence of a value for that parameter at a certain location in the plant. If the parameter is used by the rules in the ASPX knowledge base, the parameter is called a required parameter. The parameters which are solely recorded for filing purposes are optional parameters. For the development of ASPX, the data base records beginning in 1987 are obtained and converted.

The original data base records were stored in a Macintosh format where the operators used the computers solely for file keeping purposes. The data records are first transferred to the IBM format, and then to the ASCII text file format. The conversion of the old data records required an external data base conversion program which is written in Turbo Pascal language. This external DOS program is used to convert the ASCII data

Table 3-2: ASPX data base parameters: Type of measurements, the number of locations, frequency of measurement, and whether the parameter is required or optional

Parameter	Measurement				Required/ Optional R/O
	Field	Laboratory	Location	Frequency	
Flowrate	✓		5 locations	3 times/day	R
Return AS flowrate	✓		2 locations	3 times/day	R
Temperature	✓		6 locations	3 times/day	R
pH	✓		6 locations	3 times/day	R
Sulfide		✓	6 locations	3 times/day	O
Sulfite		✓	6 locations	3 times/day	O
Phenol		✓	6 locations	3 times/day	O
Oil & grease		✓	6 locations	3 times/day	O
TOC		✓	6 locations	3 times/day	R
TSS		✓	6 locations	3 times/day	R
PO4		✓	4 locations	3 times/day	R
NH3		✓	6 locations	3 times/day	R
NO2		✓	2 locations	3 times/day	R
NO3		✓	2 locations	3 times/day	R
Polymer	✓		2 locations	3 times/day	O
MLSS		✓	2 locations	1 time/day	R
DO	✓		2 locations	3 times/day	R
SVI		✓	2 locations	1 time/day	O
SBD	✓		2 locations	3 times/day	R
Waste sludge flowrate	✓		2 locations	3 times/day	R
Waste sludge MLSS	✓		2 locations	1 time/day	R
DAF Recycle	✓		2 locations	3 times/day	O

records to the new data base format, the Turbo Pascal's *Record Type*. The advantages of using the "record type" is that the size of one line of text may exceed 254 characters which is the maximum number of characters which can be used in the ASCII files, and the new format generates much smaller size data base files than the ASCII format.

The advantage of exceeding the 254 barrier is to significantly reduce the access time by introducing the ability of reading all the necessary data with one function. Each line in a file contains the date and time of the record, and the total of 84 values which are measured during the specified date and time. Furthermore, the elimination of the 254 barrier allows production of compact size data base files. One year's worth of data can be collected under one separate record file in a compressed form to provide portability and reduced storage requirements. One year's data occupies only half the space in a 1.44 MB diskette. Moreover, the compact size files are suitable for copying, storing, transferring, and conversion processes which are essential functions for maintaining a software/hardware integrity.

ASPX provides a wide range of data base manipulation functions including adding and changing the data base records, pattern analysis, graphics, statistics, transfer of data base files, and confirmation of the data base records. Adding/changing the data base records is the basic function of ASPX which enables the operator to keep a record of the historical information. The operator can simultaneously view up to 15 parameters of the plant process parameters. The data base entry form is designed as

a 123-like (rows/columns) fashion so that the operators can enter the values directly into the "cells."

The pattern analysis and the graphics provide a visual access to the plant's parameters. The ASPX is capable of reviewing the data up to one year ($365 \times 3 = 1095$ data points) to establish a pattern of the parameter trend. ASPX further provides an option to view the data trends in daily, weekly, monthly, or yearly cycles. The graphical feature is designed to provide information on the variable interrelations and the parameter tracking in the plant processes. The variable interrelation is used to understand the effect of one parameter on the other. For example, the variable interrelationship may be used to explain the effect of the dissolved oxygen concentration and the nitrification process. The parameter tracking, furthermore, provides a valuable access to knowledge of how the parameter is influenced by the individual unit processes. For example, the operator can review how the COD decreases along the plant flow to identify the COD treatment scheme.

The data base statistics supply invaluable historical information about the parameter which can be used during the expert systems consultation sessions to provide a basis for the findings. The statistical information may also be used for understanding the parameter value. The values declared by the statistics include the typical value, value desired now, value considered abnormal, and the number of times the value is considered abnormal within the last 3-days.

The ASPX data base renders functions for conversion of the records to the ASCII format which enables the data transfer to the worksheet programs. If the user desires to use external worksheet programs, the data records may be converted to the ASCII format which then can be transferred to the other programs by their file manipulation functions. Popular worksheet programs such as Lotus 1-2-3®, Excel®, and many others are capable of reading the ASCII format files to provide a wide range of possibilities with the data base record manipulation.

The backup option in ASPX is a safeguard against system crash. The data base files may be duplicated in floppy diskettes for the protection of the data. The frustration of losing data can be overcome by regular practice of backing up the data base files at least every 4 days. The internal backup-check function in ASPX warns the user in case the data base files have not been backed up for at least 5 days. The warning signal is an annoying characteristic of ASPX until its value is understood following the system crash.

The data base records may be confirmed through the verification function within ASPX which provides both checks for missing and incorrectly recorded data. The missing data is a result of not recording the value into the data base. And incorrect data recording results from a typo, or a malfunctioning equipment read out. The missing values are declared by an "X," or an "Req'd" symbol depending on whether the value is an optional or a required parameter, respectively. The check for incorrect values is performed by comparing the typical range of a parameter with

the recorded value. If the recorded value is out of this default range, the user is warned about possible error in the recording. The user has a choice of disregarding the value, or accepting the value as it is.

One of the most important functions of the data base is the statistical and the mathematical models. The models supply additional information to the ASPX knowledge base input by incorporating the real-time values into forms which the knowledge base can use. The role of the models in the knowledge base is to build a bridge between the theoretical knowledge and the selected refinery treatment plant. The models provide a customized version of the knowledge base even though the rules are applicable to other treatment plant processes. The limits which are used by the knowledge base rules are updated by the data base functions to generate the results more realistically. The models are discussed in the following sections in detail.

3.1.5.1 ASPX Statistical Model

The use of statistical models enables the expert system program to access the plant's historical characteristics. Based on the analysis of the historical data base, invaluable information is obtained in order to understand the behavior of the system. In this dissertation, the statistical model is used to determine the fuzzy terms and to predict the minimum MCRT value which produces the effluent ammonia concentration below what is permitted by the NPDES.

Determining the fuzzy terms. The first function of the statistical model is to determine the fuzzy terms used in the knowledge base rules such as "low," "high," and "normal." These terms are frequently used within the knowledge base rules because they give flexibility to the knowledge base by allowing changes in the parameters without any need for modification to the actual rules. The values for the fuzzy terms are transferred into the knowledge base through the input text files. Before the expert system is triggered, the statistical model runs and writes the results in a text file which then is imported into the values in the rules during the expert system consultation.

The parameters which have fuzzy terms are listed in Table 3-5 located in section 3.1.6. For each of these parameters, the fuzzy terms must be declared in order for the expert system to function properly. The default fuzzy terms are declared by the developer and may be altered by an authorized supervisor to meet the needs of that specific plant. The default values represent values taken from literature. Although these values are characteristic of most plants, alteration of these limits may be required for each specific plant.

The statistical model assigns values for the fuzzy terms based on the analysis of the historical data base. For example, for the rule shown below, the value for "DO Low" must be declared in order to evaluate the rule.

IF DO < DO Low
THEN ADVICE= Open air discharge valve

To determine the value for "DO Low," the statistical model reads the DO values from the preceding two months to the current date. The model performs the frequency analysis on the data read from the data base. For example, based on this analysis, the model determines the specific frequency of having different DO levels in the aeration basins during the time period covered. The frequency graph is generated upon a request from the user to show the distribution of the DO levels. Figure 3-6 shows an example case of the frequency distribution for the DO concentration in the aeration basins.

The fuzzy terms are determined based on the result of the frequency analysis. The upper and lower 5% of the frequency represents the high and low value, respectively. The median value shows the most frequently observed value for that parameter. For the above example, the low limit for the DO concentration can be determined from Figure 3-6 as 0.5 mg/L. Moreover, the high value for the DO is determined to be 3.6 mg/L and the median value equals 1.6 mg/L.

The 5% limit is an arbitrary number and changes for each plant. Since there is no defined method to pick the limit values, for the purpose of this dissertation the 5% limit represents an adequate range to detect the high and low values.

If the calculated fuzzy term from the frequency analysis does not match the default value declared by the developer, the expert system program notifies the operator to verify the conflicting values. In some cases, the default

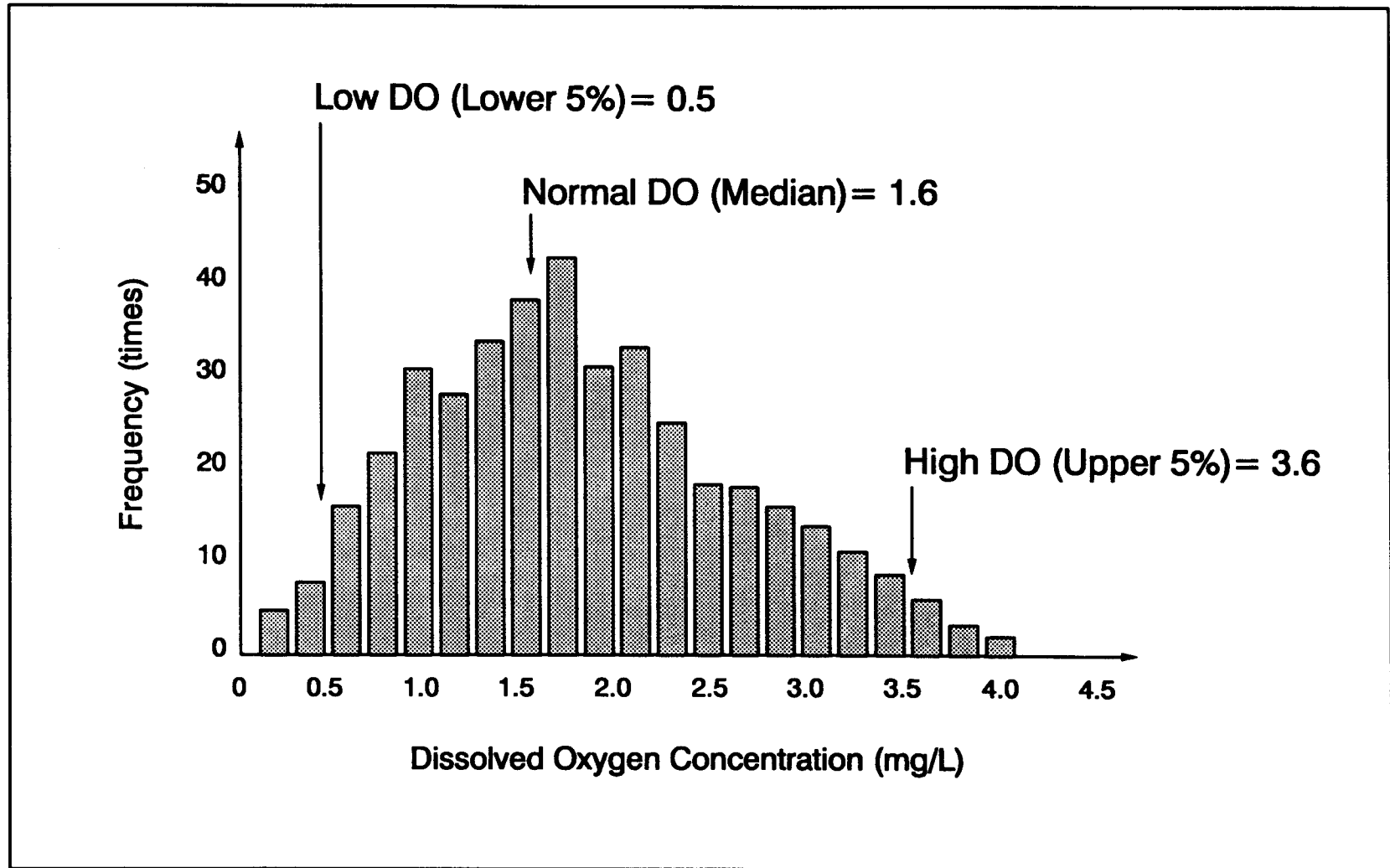


Figure 3-6: Example frequency analysis for the dissolved oxygen with the high, low and normal values

value may need to be adjusted or an error might be detected in the data base records.

Determination of the fuzzy terms enables the plant to continuously adapt to changes by providing a customized program for the selected treatment plant. Furthermore, since the fuzzy terms reflect past experiences with the plant, this will enable more reliable real-time operations of the plant.

Prediction of the minimum MCRT. The second function of the statistical model in this dissertation is to predict the minimum MCRT to achieve an effluent ammonia concentration below the NPDES permit. The Bayesian statistical method is used to predict the MCRT. An overview of Bayesian statistics is given in section 2.2.3. The function of Bayesian statistics is to predict the probability of having the effluent ammonia concentration below the desired level for a given range of MCRT values.

In the application of the Bayesian method, the *State* variable is defined as the effluent ammonia concentration. Two state range are defined as follows:

$$\text{State}_1 = S_{\text{NH},e} < (0.75) \text{ NPDES}$$

$$\text{State}_2 = S_{\text{NH},e} \geq (0.75) \text{ NPDES}$$

Seventy-five percent (75%) of the NPDES is chosen for the range of the state variable because there are not enough observations made above the NPDES permit as this would mean a violation of the permit. To obtain statistically feasible results, the upper limit was lowered to 75% of the

ammonia concentration permitted by the NPDES. The $State_1$ variable indicates that there is no violation of the NPDES permit. Although the $State_2$ variable is evaluated based on 75% of the NPDES, it is assumed that it shows a violation of the NPDES permit.

For the *Sample* variable, the MCRT was chosen because it is the key process control parameter in controlling the nitrification process and the minimum MCRT value needed to be determined. The following six *Sample* ranges are identified for the MCRT values (in days):

$$Sample_1 = MCRT < 5$$

$$Sample_2 = 5 \leq MCRT < 10$$

$$Sample_3 = 10 \leq MCRT < 18$$

$$Sample_4 = 18 \leq MCRT < 25$$

$$Sample_5 = 25 \leq MCRT \leq 35$$

$$Sample_6 = MCRT > 35$$

Under these sample variable ranges and the two state conditions declared above, the Bayesian statistics is able to create a table of probabilities by using Eq. 2-53. A sample table is shown in Table 3-3 where the probability of a violation and no-violation is declared based on the sample variable ranges. For example, if the MCRT is between 10 to 18 days ($Sample_3$), there is a 25% chance that there will be a violation ($State_1$) of the NPDES permit.

Table 3-3: Example of the Bayesian statistical table for determining the minimum MCRT value

MCRT (days)	<u>Violation</u> State ₁	<u>No Violation</u> State ₂
MCRT < 5 (Sample ₁)	0.95	0.05
5 ≤ MCRT < 10 (Sample ₂)	0.75	0.25
10 ≤ MCRT < 18 (Sample ₃)	0.25	0.75
18 ≤ MCRT < 25 (Sample ₄)	0.09	0.91
25 ≤ MCRT < 35 (Sample ₅)	0.06	0.94
MCRT > 35 (Sample ₆)	0.06	0.94

The minimum MCRT value is determined based on the following criteria. The minimum value for the MCRT is equal to the lowest MCRT range in the Bayesian table which indicates a 90% or higher chance of a no-violation state (State₂). For example, in Table 3-3, the minimum MCRT value is determined as the Sample₄ range because it is predicted that if the MCRT is between 18 and 25 days, there is a 91% chance that there will not be a violation of the permit. Although MCRT ranges above 25 days (Sample₅ and Sample₆) are predicted to have lesser chance of violation (6%), they are not chosen because maintaining a lower MCRT (18 to 25 days) would produce an adequate treatment efficiency. Furthermore, it is assumed that the selection criteria of 90% is statistically feasible for the purpose of this dissertation.

If none of the probabilities of State₂ qualify for the 90% selection criteria, then the range which predicts the highest probability is chosen as the minimum MCRT value. Although theoretically, the highest probability

should be observed for the highest MCRT value ($State_6$), it is not always true for real-time operations. Therefore, the use of the Bayesian statistical method would provide a more realistic representation of the system's behavior based on past experiences.

There are a number of "pitfalls" needed to be checked when using the Bayesian statistics. In particular the presence of an adequate number of data points for a given sample or a state variable must be checked. If there is not an adequate number of data points present for the Bayesian statistics to accurately predict the future, the operator is warned about the uncertainty of the results. Furthermore, additional warning messages are displayed if all the sample ranges fail during the 90% selection process or any probability predicted by the Bayesian statistics equals "zero."

3.1.5.2 ASPX Mathematical Model

The mathematical model for nitrification consists of kinetic reactions of both *Nitrosomonas* and *Nitrobacter*. Some of the mathematical models developed for the activated sludge process in the literature are discussed in section 2.1.5.

The primary objective of a model is to simulate the outcome of the process by analyzing the reactions taking place inside the system. Models are frequently used to estimate the future outcome of a process and to adjust the process control parameters so that the desired outcome is achieved.

For example, a model for the nitrification process can simulate the effluent ammonia concentration for the next 3 days. Furthermore, by running many simulations, an optimum pH, sludge age, or DO concentration can be estimated to achieve the desired degree of nitrification.

This dissertation combines the mathematical models and the expert systems to benefit from the advantages offered by these two modules and to eliminate their weaknesses by incorporating the two modules together. For the purpose of this dissertation, a moderately complex model would be adequate since more complex models may take extended computer time for the solution and they do not provide more accurate results.

The mathematical model adopted by this dissertation is used to simulate the effluent ammonia concentration and to estimate the minimum sludge age required to achieve a nitrification that produces effluent ammonia concentration below the NPDES permit requirements. The model is also used for training the plant operators by simulating the effluent ammonia concentration based on the adjustments made on pH, temperature, sludge age and the dissolved oxygen concentration. Because the model output is not used directly in the decision making process, only an estimation is requested from the model. Moreover, the solution time for the model equations is reduced because a moderately complex model is applied into the ASPX structure.

The basic structure of the model for this dissertation is adopted from Poduska and Andrews (1975). The model was derived based on the

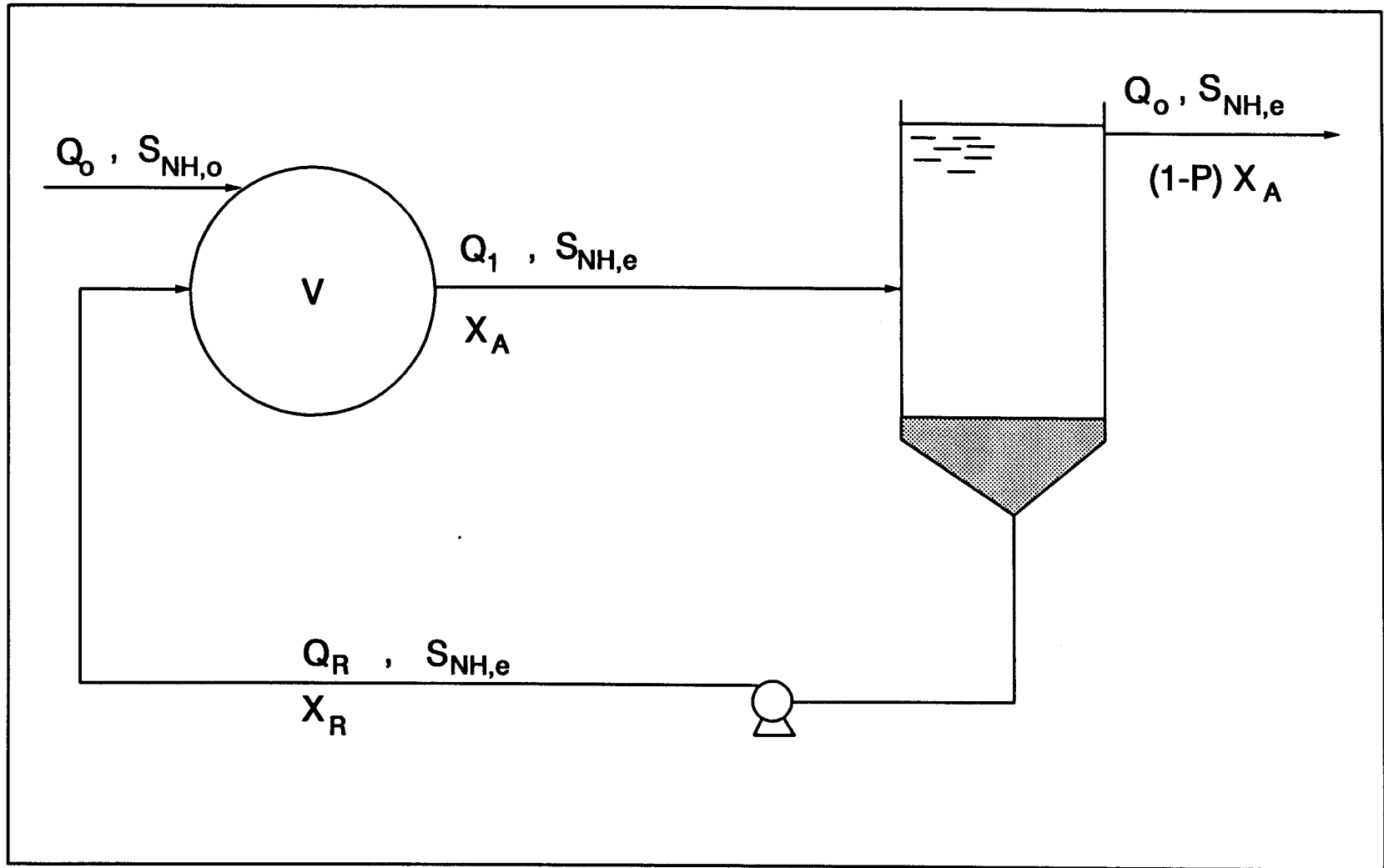


Figure 3-7: Schematic diagram of the reactor setup for the model developed by Poduska and Andrews (1975)

reactor system shown in Figure 3-7. The general differential equations were developed based on the mass balance around the system. For the continuous flow stirred-tank reactor, the rate of substrate utilization and the rate of microbial mass change were estimated based on the following general formulation:

$$\left[\begin{array}{c} \text{Rate of} \\ \text{Accumulation} \end{array} \right] = [In] - [Out] \pm \left[\begin{array}{c} \text{Utilization} \\ \text{or} \\ \text{generation} \end{array} \right] \quad (3-1)$$

Although the model is developed for ammonia, nitrite, and nitrate, for the purpose of this dissertation, only the ammonia utilization equations are adopted. In the model, the ammonia concentration change was formulated based on the substrate material balance considering Eq. 3-1. The mass balance for the ammonia nitrogen was identified as:

$$V \frac{dS_{NH,e}}{dt} = Q_o S_{NH,o} - Q_e S_{NH,e} - V \mu_A \frac{X_A}{Y_A} \quad (3-2)$$

where $S_{NH,e}$ = Effluent NH_4^+ -N concentration, mg/L

Q_o = Influent flowrate, gpd

$S_{NH,o}$ = Influent NH_4^+ -N concentration, mg/L

The growth equations based on the mass balance for both *Nitrosomonas* and *Nitrobacter* were also created in the model. Because the most significant factor in the nitrification process is the growth of *Nitrosomonas*, only the growth equation for *Nitrosomonas* is taken from the model. The mass balance for the autotrophic bacteria mass was determined as follows:

$$V \frac{dX_A}{dt} = -Q_o X_A (1.0 - P) + V X_A (\mu_A - b_A) \quad (3-3)$$

where P = Efficiency of solid-liquid separation

After the basic model is adopted from Poduska and Andrews (1975), further alterations are made to account for the effects of DO, pH, temperature, and the ammonia inhibition factor. The effect of DO is significant in the nitrification process (see section 2.1.2.3). The DO concentration becomes an additional substrate limiting factor when the DO levels are below 1.0 mg/L. The basic expression to indicate the DO effects on the nitrification kinetics is adapted from Stenstrom and Poduska (1980). The expression is modified to include the ammonia inhibition term as shown below:

$$\mu'_A = \mu_{m,A} \left[\frac{S_{NH}}{K_{NH} + S_{NH} + (S_{NH})^2 / K_i} \cdot \frac{S_o}{K_{o,A} + S_o} \right] - b_A \quad (3-4)$$

In Eq. 3-4, the specific growth rate ($\mu_{m,A}$) is estimated by the correlation equation proposed by Antoniou *et al.* (1990) based on the pH and the temperature of the system. As it is discussed in section 2.1.2.3, the effect of pH and temperature is a significant factor in the nitrification kinetic equations. The proposed correlation between $\mu_{m,A}$, pH and temperature is used to predict the value of $\mu_{m,A}$ for a given pH and temperature values as shown below (same as Eq. 2-29):

$$\mu_{m,A} - b_A = \frac{m e^{-(a/T)}}{1 + \frac{b}{10^{-pH}} + \frac{10^{-pH}}{c}} \quad (3-5)$$

$$= A$$

Rearranging:

$$\mu_{m,A} = A + b_A \quad (3-6)$$

Applying Eq. 3-6 into 3-4:

$$\mu'_A = (A + b_A) \left[\frac{S_{NH}}{K_{NH} + S_{NH} + (S_{NH})^2/K_i} \cdot \frac{S_o}{K_{o,A} + S_o} \right] - b_A \quad (3-7)$$

From the definition of the net specific growth rate:

$$\mu_A = \mu'_A + b_A \quad (3-8)$$

Combining Eq. 3-2, 3-7, and 3-8:

$$V \frac{dS_{NH,e}}{dt} = Q_o S_{NH,o} - Q_e S_{NH,e} \quad (3-9)$$

$$- V \frac{X_A}{Y_A} \left(\frac{m e^{-(a/T)}}{1 + \frac{b}{10^{-pH}} + \frac{10^{-pH}}{c}} + b_A \right) \frac{S_{NH,e}}{K_{NH} + S_{NH,e} + (S_{NH,e})^2/K_i} \cdot \frac{S_o}{K_{o,A} + S_o}$$

Similar to the substrate equation, the *Nitrosomonas* mass balance equation is altered by replacing the term $(\mu_A - b_A)$ in Eq. 3-3 with the term μ'_A in Eq. 3-7 based on the relationship given in Eq. 3-8. The change in the *Nitrosomonas* mass thus is formulated as follows:

$$\begin{aligned}
V \frac{dX_A}{dt} = & Q_o X_A (1.0 - P) \\
& + V X_A \left[\left(\frac{m e^{-(a/T)}}{1 + \frac{b}{10^{-PH}} + \frac{10^{-PH}}{c}} + b_A \right) \frac{S_{NH,e}}{K_{NH} + S_{NH,e} + (S_{NH,e})^2/K_i} \cdot \frac{S_o}{K_{o,A} + S_o} - b_A \right]
\end{aligned}
\tag{3-10}$$

Eq. 3-9 and 3-10 are solved numerically by using the Fourth Order Runge-Kutta method (for details of the method, see James *et al.*, 1985). The values for the coefficients in Eq. 3-9 and 3-10 are given in Table 3-4. The kinetic constant values are determined based on the values reported in the literature for the activated sludge operating above 20°C (see Table 2-10b and Table 2-13).

In addition to the declaration of the values for the parameters, the initial values for the parameters are also needed to solve Eq 3-9 and 3-10. All the initial values are directly read from the previous day's record in the data base with the exception for X_A . The initial X_A ($X_{A,0}$) is estimated by either the BOD₅/TKN ratio method (see section 2.1.2.3) or by solving Eq. 3-9 for $X_{A,0}$ assuming the steady state conditions ($dS_{NH,e}/dt = 0$). The estimation by the BOD₅/TKN ratio method is performed at the beginning of the simulation, by calculating the fraction of the nitrifiers (f) using Eq. 2-26. The initial value for X_A is estimated by the following equation:

$$X_{A,0} = (f) (X_{H,VSS}) \tag{3-11}$$

where $X_{H,VSS}$ = VSS fraction of the heterotrophic biomass, mg/L

Table 3-4: Values used for the constants in Eq. 3-9 and 3-10

Parameter	Value used in the model
m	4.70×10^{14}
a	9.98×10^3
b	2.05×10^{-9}
c	1.66×10^{-7}
Y_A mg VSS/ mg $\text{NH}_4^+\text{-N}$	0.25
b_A d^{-1}	0.10
K_{NH} mg $\text{NH}_4^+\text{-N/L}$	1.50
K_i mg $\text{NH}_4^+\text{-N/L}$	250.00
$K_{o,A}$ mg/L	0.50

From the steady state solution of Eq. 3-9, the initial X_A is also estimated by the following equation:

$$X_{A,o} = \frac{Y_A}{V} \cdot \frac{(Q_o S_{\text{NH},o} - Q_o S_{\text{NH},e})}{(A + b_A) \cdot \frac{S_{\text{NH},e}}{K_{\text{NH}} + S_{\text{NH},e} + (S_{\text{NH},e})^2/K_i} \cdot \frac{s_o}{K_{o,A} + S_o}} \quad (3-12)$$

The initial X_A equals to the value estimated by either Eq. 3-11 or 3-12, whichever is the lowest.

The model is used for two functions, the simulation of the effluent ammonia concentration and the determination of the minimum MCRT to achieve a desired effluent ammonia concentration.

Simulation. The model is used to estimate the effluent ammonia concentration over a given time period. The simulation is performed by solving Eq. 3-9 and 3-10 for a period of 3 days assuming the current conditions including pH, DO, and temperature will stay the same during the simulated time period. A graph such as the one shown in Figure 3-8 is displayed to the operators from which they can see the future outcome of the system as it is predicted by the model.

For the numerical solution of Eq. 3-9 and 3-10, an optimum time step (Δt) is determined by considering the accuracy, the stability of the solution technique, and the computational time requirements. After performing several runs with the simulation program, the optimum time step is determined to be 0.01 d.

The simulation of the effluent ammonia concentration provides invaluable information to the expert system program. At the end of the simulation, if the predicted trend exceeds the NPDES permit, a signal is sent to the expert system program to notify the operator about the possible violations in the future. The advice given by the expert system is different based on whether the violation will occur or the effluent ammonia will stay below the NPDES permit.

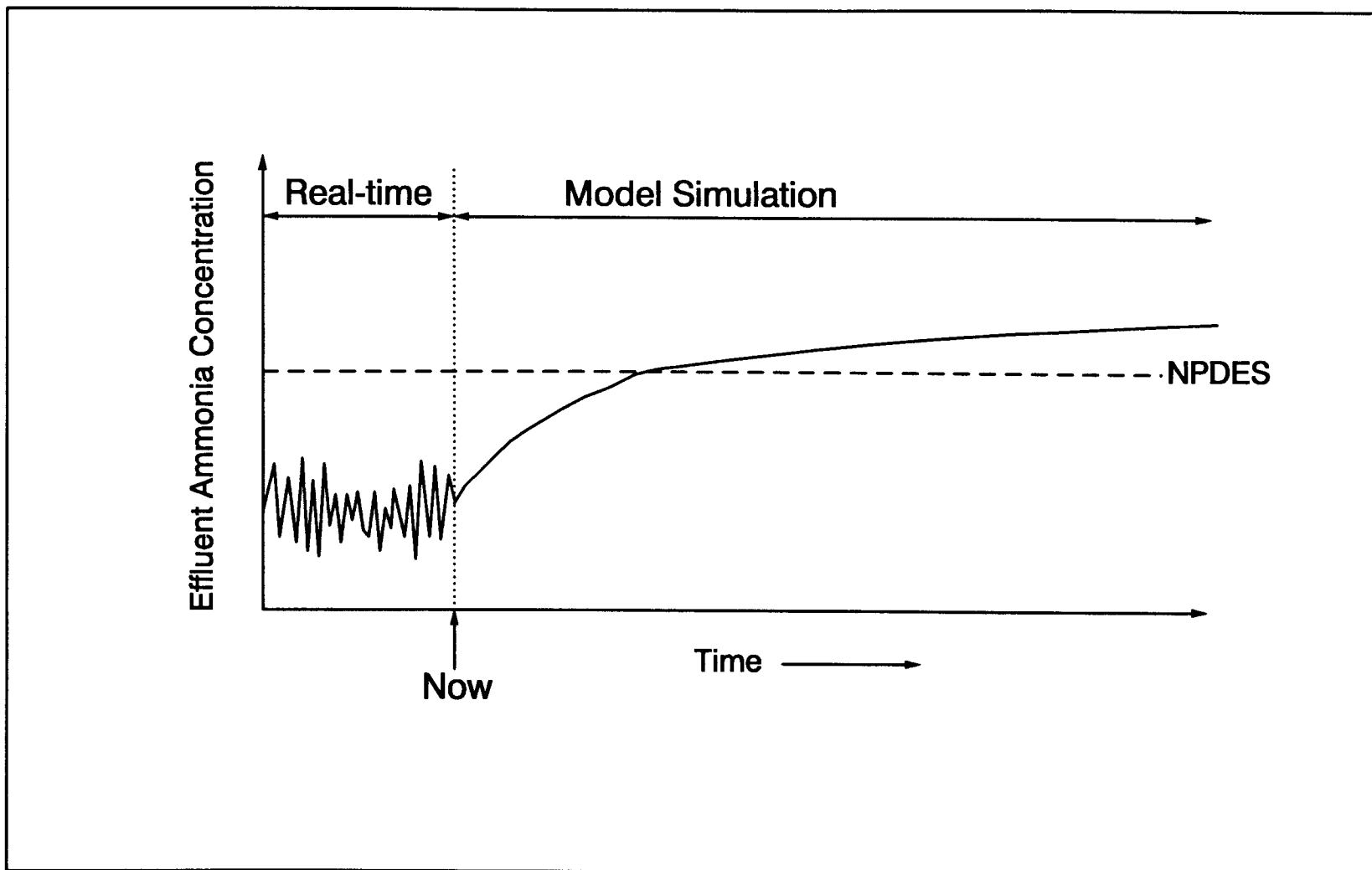


Figure 3-8: A sample mathematical model simulation result to predict the effluent ammonia concentration for the next 3 days

Similar information about the effluent ammonia concentrations is provided by the frequency analysis within the statistical model explained in section 3.1.5.1. The difference between the mathematical and the statistical model is that the statistical model declares whether the effluent ammonia concentration is considered low, high, or normal while the mathematical model predicts a numerical value for the effluent ammonia concentration. Both informational forms, however, are an invaluable part of the expert system consultation.

The simulation model for the effluent ammonia concentration is also an effective tool for educating the operators on the nitrification process. The operators change pH, DO, temperature, and MCRT values and observe the future outcome based on the calculations made by the model. In this way, the effect of each parameter is understood after several tries with the simulation program.

Determining the minimum MCRT. The model is also used to determine the minimum MCRT which provides an effluent ammonia concentration below the desired levels. Although more complex models may optimize MCRT, pH, and DO together, within the scope of this dissertation, it is assumed that the process control of MCRT is the key parameter in the nitrification process.

MCRT is incorporated into the differential equations based on the definition of MCRT as shown below:

$$MCRT = \frac{V X_A}{Q_o X_A (1 - P)} \quad (3-13)$$

$$P = 1 - \frac{V}{Q_o MCRT} \quad (3-14)$$

To determine the minimum MCRT which produces the effluent ammonia below the given level, Eq. 3-9, 3-10 and 3-14 are solved for the MCRT range of 2 to 30 days with the 1 day increments of the MCRT value. A new value for the parameter P is calculated using Eq. 3-14 for each MCRT step. After calculating $X_{A,o}$, X_A is next calculated by Eq. 3-10 for the new P value. The effluent ammonia concentration is then calculated by Eq. 3-9 for given X_A and MCRT at the end of one simulation day ($t= 1$). At the end of the simulation, a graph such as the one shown in Figure 3-9 is generated for the operator to review the results of the model calculations.

The minimum MCRT is equal to the lowest MCRT which produces an effluent ammonia concentration below the NPDES permit (see Figure 3-9). The minimum MCRT represents the lowest MCRT allowed in order to achieve the desired treatment efficiency. Once the minimum MCRT is determined, this information is transferred to the expert system program where it is further evaluated.

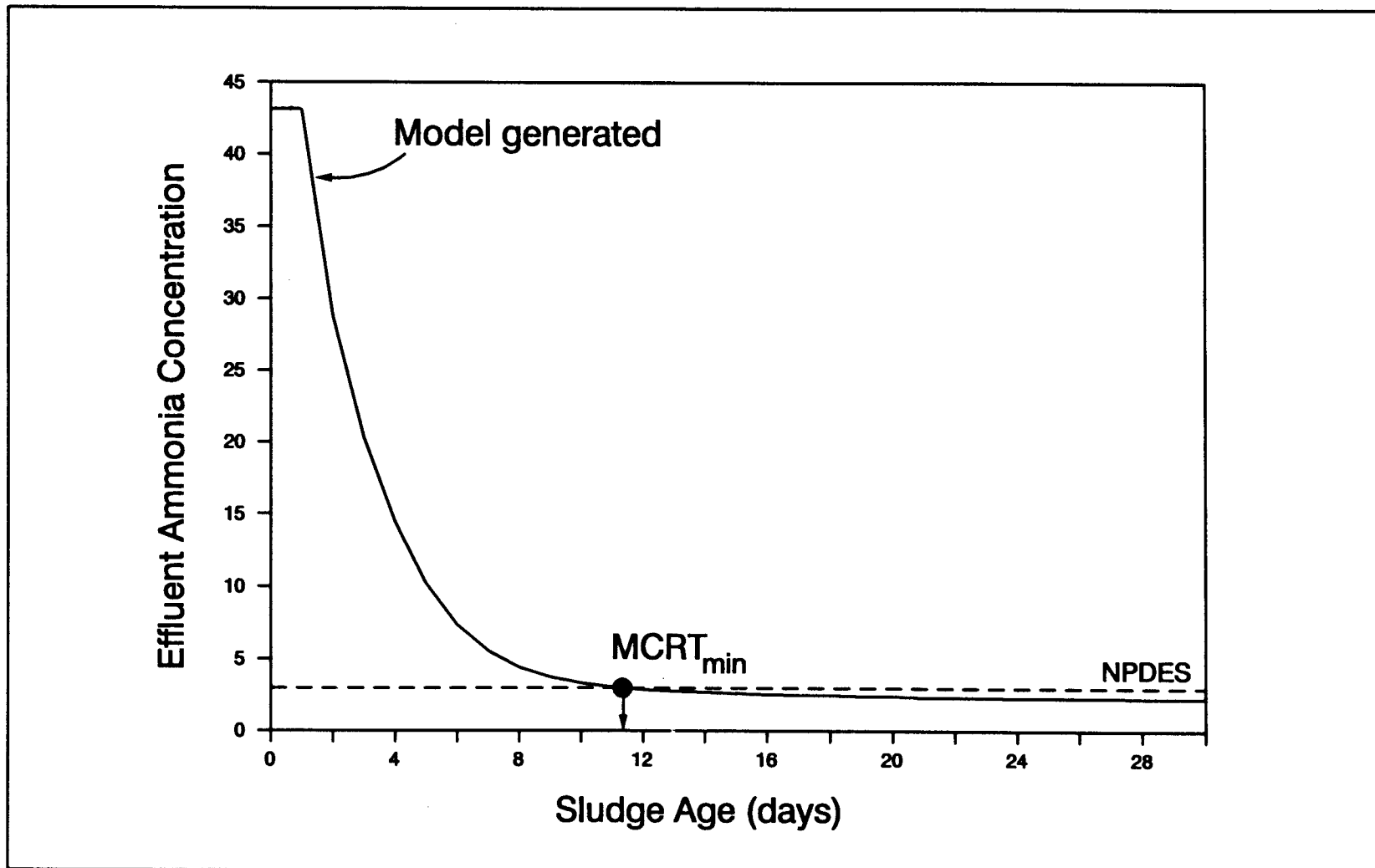


Figure 3-9: Determination of the minimum MCRT value as it is calculated by the mathematical model

A similar prediction is made by the statistical model by observing the real-time plant data. Although the statistical model does not give an exact value for the minimum MCRT, it gives a likelihood of having a certain range of MCRT based on the historical data base values. The information from both the statistical and the mathematical model is transferred to the input files for the expert system program. During the expert system consultation, the operator is advised to adjust the MCRT value according to the estimation made by the models.

3.1.6 ASPX Expert System

Prior to consultation, the parameters which are used in the rules must have declared limit values so the inference engine can evaluate fuzzy terms such as low and high. In addition to the declaration of parameter limits, the states which determine the degree of nitrification loss must be defined to ensure the integrity of answers given by ASPX.

The ASPX's knowledge base consists of 337 rules, and 56 parameters which are divided into seven sub-frames. Within the knowledge base, fuzzy terms such as "low," "high," and "okay" are used to avoid rigid rules. During the knowledge base evaluation process, these fuzzy terms are defined based on given individual limiting criteria (fuzzy terms) used for each parameter. These limiting criteria determine whether the parameter has an acceptable (okay) or an abnormal value (low or high).

Table 3-5: Parameters that have limiting values in the ASPX program and the default values chosen for the selected refinery plant

Influent parameters	Influent pH (Allowed range)	3.2 - 8.5	
	Organic loading (High limit)	3,100	kg/d
	Influent NH ₄ ⁺ -N (High limit)	100	mg N/L
	Influent temperature (Low limit)	70	°F
	Influent phenol (High limit)	15	mg/L
	Influent sulfide (High limit)	50	mg/L
Activated sludge parameters	pH level (Low limit)	6.8	
	Dissolved oxygen (Low limit)	3.5	mg/L
	MLSS (Low limit)	3,000	mg/L
	MLSS (Safe limit)	4,500	mg/L
	F/M ratio (Low limit)	0.05	
	F/M ratio (Optimum value)	0.15	
	MCRT (Desired value)	21	days
	OUR test result (Low limit)	10	mgDO/L·d
	Phosphoric acid addition (Low limit)	10	gpd
	Caustic addition (Low limit)	30	gpd
	Temperature (Low limit)	78	°F
Blowers	Air flowrate (Low limit)	20	SCFM
	Suction pressure (High limit)	50	psi
	Discharge pressure (Low limit)	300	psi
	In-take filter replacement (Optimum)	3	mths
	Diffuser cleaning (Optimum value)	6	mths
Clarifier tank parameters	WAS flowrate (Optimum value)	15	gpd
	WAS MLSS (Low limit)	7,000	mg/L
	RAS rate (Low limit)	40	% Q
	Sludge blanket depth (High limit)	3	ft
Effluent parameters	Phosphate (Low limit)	10	mg/L
	Ammonia-N (High limit)	5	mg N/L
	TOC (High limit)	40	mg/L
	TSS (High limit)	50	mg/L

Table 3-5 shows the parameters that have limiting values, and the default values which are taken from literature for those parameters. To maintain the integrity of the program, the limits may only be changed by a supervisor under restricted access for customizing purposes only.

During the expert system consultation, the inference engine evaluates the rules based on the search techniques explained in section 2.2.2.6. Depending on the state of the plant, the inference engine searches for answers by initiating one or more sub-frames within the ASPX's knowledge base. The rules are evaluated and suggestions are given based on the current state of the nitrification process in the plant.

In the selected plant, four states of the nitrification process are defined to distinguish the differences between the severity of the nitrification failures. The defined four states are as follows: *severe problem* (no nitrification is occurring); *problem* (some nitrification is occurring); *potential problem* (nitrification will be lost within the next 2-3 days); and *optimum operation* (nitrification is occurring). The states are determined by fitting a straight line through the last 5 days' effluent ammonia concentration by using the Least-squares (James *et al.*, 1985) method. Three factors, the slope of the line, the last day value, and the total value within the last 5 days determine the plant's current nitrification state. The maximum allowed variance within the actual data is also examined to detect possible error in measuring equipment. Figure 3-10 illustrates the factors that affect determination of the nitrification state.

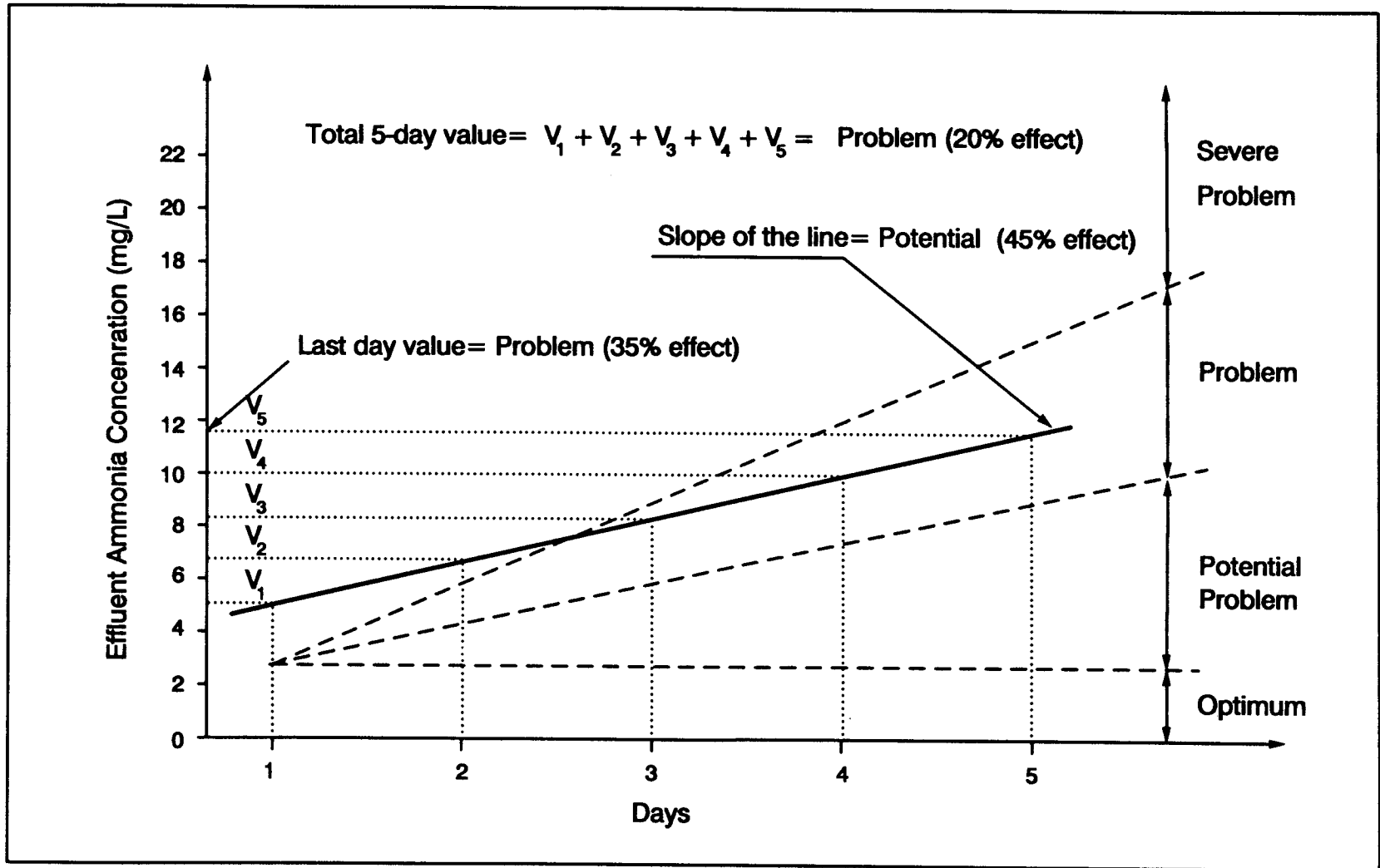


Figure 3-10: An example of determining the nitrification state

These three factors differ in their impact in determining the nitrification state. The weighing factors are assigned for each of these three factors to indicate the degree of importance in the nitrification state determination. For example, the slope of the line has 45% effect, the last day value has 35% effect, and the total 5-day value has 20% effect on the determination process.

Total weighing factors are equal to 100 to indicate the percent of effect of each factor. The state values are also declared to determine the nitrification state. Once the line is fitted through the last 5 days' effluent ammonia data, the three factors and corresponding state value are manipulated to obtain final state value. This final state value determines the current nitrification state of the plant. The default state values are entered into ASPX but they can be changed for customizing purposes if necessary.

Figure 3-10 gives an example of how the nitrification state is determined. In the determination process the state value is calculated by multiplication of the weighing factor and the corresponding state value. In Figure 3-10, the slope of the fitted line equals "Potential Problem" slope, the last day value equals "Problem" value, and the total 5-day value equals "Problem" value. Using the weighing factor the combination of "Potential," "Problem," and "Problem" is determined to be "Problem." The program code to determine the nitrification state is included in Appendix D.

Based on the current nitrification state, the expert system's advice is modified to ensure the consistency between the plant's nitrification state and the given advice. For example, if the detected problem is low pH, the advice would be to increase the amount of caustic when the nitrification state indicates "potential" problem. The advice would have been to check the pH electrode and add emergency caustic solution if the nitrification state indicated "severe" problem.

The operators may evaluate two activated sludge trains in the plant separately by three different techniques during the expert system consultation. Figure 3-11 shows the dialog box displayed by the user-interface program to run the expert system shell with different options. The operator either selects the ASP 600 train or the ASP 700 train which both are trains in the selected refinery plant. Then the operator has a choice of "limited," "regular," or "in-depth" analysis techniques provided by the expert system (see Figure 3-11).

The limited technique asks no questions of the user during the consultation session and only detects the abnormal values recorded in the data base records, and the most obvious abnormality within the plant. The limited analysis provide a quick look at the plant processes. The operator uses this option regularly to detect problems fast and without providing extensive information into ASPX. This version of analysis is the quickest way to find the most apparent nitrification problem in the plant.

Data Base	Expert Systems	Reports	ASPX Activated Sludge Process eXpert Produced by: Naci Ozgur							
<div style="border: 1px solid black; padding: 5px; margin-bottom: 10px;"> Run diagnosis Review results Set limits </div> <div style="border: 1px solid black; padding: 10px; background-color: #e0e0e0;"> <p style="text-align: center;">Run Diagnosis</p> <table style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th style="text-align: left; border-bottom: 1px solid black;">Train</th> <th style="text-align: left; border-bottom: 1px solid black;">Technique</th> </tr> </thead> <tbody> <tr> <td><input checked="" type="checkbox"/> ASP 600 & 601</td> <td><input type="checkbox"/> Limited</td> </tr> <tr> <td><input type="checkbox"/> ASP 700 & 701</td> <td><input checked="" type="checkbox"/> Regular</td> </tr> <tr> <td></td> <td><input type="checkbox"/> In-depth</td> </tr> </tbody> </table> <p style="text-align: center; margin-top: 10px;">RUN</p> </div>				Train	Technique	<input checked="" type="checkbox"/> ASP 600 & 601	<input type="checkbox"/> Limited	<input type="checkbox"/> ASP 700 & 701	<input checked="" type="checkbox"/> Regular	
Train	Technique									
<input checked="" type="checkbox"/> ASP 600 & 601	<input type="checkbox"/> Limited									
<input type="checkbox"/> ASP 700 & 701	<input checked="" type="checkbox"/> Regular									
	<input type="checkbox"/> In-depth									
Glossary	HELP		<div style="border: 1px solid black; padding: 5px; text-align: center;"> Quick Access Data base Run ASPX </div>							

Figure 3-11: The dialog box displayed by ASPX to run the expert system shell with different options

More comprehensive diagnosis may be performed by running the regular analysis. Although the expert system shell may ask as many as 10 questions during the consultation session, more detailed analysis of the plant reveals additional problems which are detected by the limited version. The operator is asked to provide additional information such as the blower valve setup, or amount of caustic addition to detect problems more accurately.

The most extensive analysis of nitrification problems can be accomplished by the in-depth analysis technique. The operator is asked to provide answers to about 25 questions which disclose all detectable problems with the nitrification process. The in-depth version is designed to be used during the "severe" problem states when there is nowhere else to turn. This comprehensive diagnosis offers invaluable solutions to problems that otherwise would not have been detected by the other two techniques. A limited diagnosis takes about 15 seconds on an 386-based PC running at 20 MHz and an in-depth diagnosis may take up to 2 minutes to complete on the same computer speed.

The results from the expert system consultation are stored in codes to allow modification of the answers given by ASPX. When the expert system shell concludes the consultation session, the resulting codes are transferred back to the user-interface program where the codes are decoded and the answers are displayed to the user. The decoding is performed within the user-interface by reading the codes from output text files and finding corresponding code numbers in the separately dedicated code data

base text file. The answer contains information on the abnormalities detected by ASPX, as well as the corrective actions which the operator should take to recover from those detected abnormalities.

The corrective actions given by ASPX clearly declare the action procedure step by step including quantitative information such as amount of caustic to add, desired setting for the blower discharge valve, or exact volume of sludge to be disposed. The quantitative information is specifically calculated for the selected plant based on the process control strategy discussed in section 4.3.

3.1.7 ASPX Report Generation Module

The report generation part of ASPX helps operators keep up with the report filing and record keeping requirements by generating the daily turnover sheets, graphical data base reports, hourly problem progress reports, and operator activity reports.

The daily turnover sheets are printed daily by ASPX for hard-copy reviewing and for record keeping purposes. A sample turnover sheet, which contains daily measured values, is shown in Figure 3-12.

The parameter values can also be printed in a graphical form to show the pattern/trend for a particular parameter. The desired duration of the graph is indicated by the operator depending on the parameter. For example,

	Feed	DAF 202	DAF 302	Eq. 500	ST 601	ST 701
Feed Q	2000	1000	1000		800	1200
Q RAS		300	100		500	750
Temp	88	85	87	88	82	83
pH	8.3	8.0	8.5	8.3	7.1	7.0
Sulfide	10	10	8	9	5	4
Phenol	5	6	4	5	1	0
IR	350	300	280	290	10	6
COD	3000	2850	2800	2500	250	230
TSS	350	320	290	310	10	5
PO4	10			15	9	10
Sulfite	15	12	13	12	0	0
NH4	90	85	83	80	2	5
NO2					10	15
NO3					65	55
Polymer		15	20			

	ASP 600	ASP 700
MLSS	3500	4250
DO	2.0	2.5
SVI	120	100

	ST 601	ST 701
SBD	3	5
Q WAS	5	0
X WAS	7500	8000

Date	05/29/91	
Time	08:00	

	I/S	S/B
K-500	X	
K-500A	X	
K-500B		X

Operators:
John Doe
Jane Roe

Supervisor:
John Doe

Figure 3-12: An example turnover sheet generated by ASPX

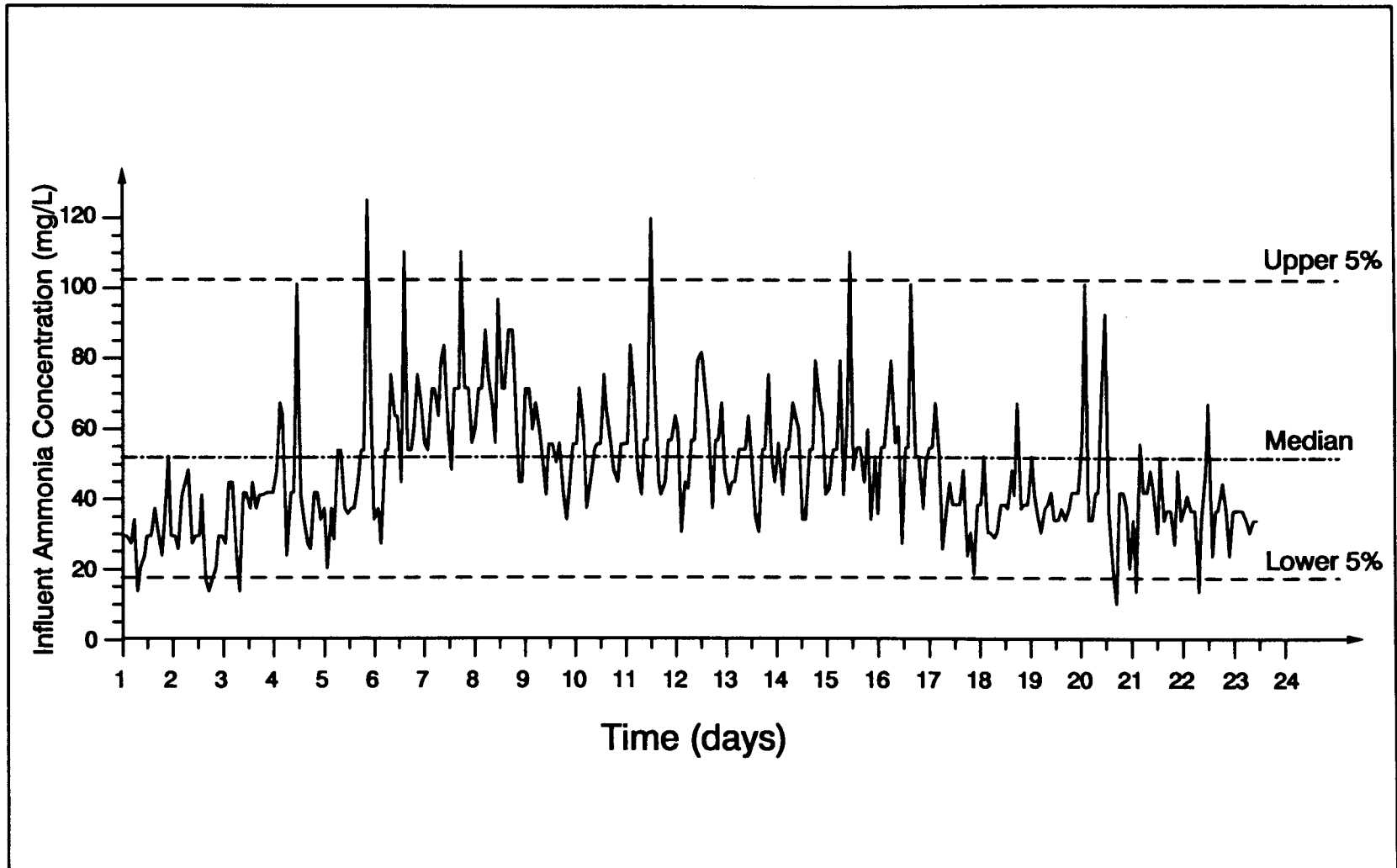


Figure 3-13: A sample graphical data base record: Influent ammonia concentration for the last 24 days with the upper and lower 5% and the median values

DO may be plotted in daily durations, while the MLSS may be reviewed for a duration of a week. The operator is able to print any desired parameter for any given duration of time. A sample graphical data base record, which shows the influent ammonia concentration for the last 24 days, is shown in Figure 3-13. Figure 3-13 also shows the upper and the lower 5% limits which designate the high and the low ammonia levels, respectively. The median line shows the approximate average influent ammonia concentration of the last 24 days.

During the problem states of the nitrification process, hourly problem progress reports may be generated to verify the actions taken to correct the problem, and to observe the effects of the problem on other unit processes in the plant. The operator is able to print follow-up reports during the problem states to inform the supervisor of the status of the problem.

The format of each report mentioned above can be changed by a supervisor to customize the format for specific use on the selected plant.

Ease of generating the reports will improve report filing procedure by the operators because ASPX will do most of the job for them. The reports generated by ASPX can be combined with the reports which the operators generate to complete the daily process control review.

3.2 Testing the ASPX Knowledge Base

Testing of ASPX is required to ensure the quality and reliability of the ASPX program. The main objective of the testing procedure is to produce a dependable, error-free, tamper-proof, and operational ASPX program that will diagnose and give advice on nitrification abnormalities in the selected plant.

The testing procedure is accomplished by the 3-step testing procedure and implementation of ASPX to the selected refinery plant. The 3-step testing procedure consists of an *initial*, *intermediate*, and *advanced* testing process. The implementation of ASPX takes place during the intermediate testing stage to obtain feed-back from site-specific evaluation of the program. The 3-step testing procedure will reveal all improvements necessary to achieve an accurate ASPX program and a highly enhanced user-interface.

Recently, the initial testing and implementation stages were completed but the intermediate and advanced testing will take place in the future and the findings of the future study will be published elsewhere. In the following sections, the procedural outline of the stages accomplished in testing, and the stages that will be achieved are given.

3.2.1 Initial Testing

The initial testing of ASPX involves verification of the knowledge base by the developer where the rules are restructured, reorganized, and rewritten to eliminate problems with the knowledge base. The testing of the rules is performed under the PC Plus development environment in which the developer can view the actions performed by the inference engine. Tracing the inference engine's actions helps the developer to locate where the problems are originating.

The PC Plus development environment offers a number of useful debugging tools to the developers during creation of a knowledge base. Some of the most useful tools are as follows: ability to trace the rule evaluation process along with the search path of the inference engine; ability to ask "why" a rule is used, and "how" a value is assigned to a parameter; and a review option which allow modifications to the responses to questions asked during consultation sessions.

The expert system developers can observe the entire rule evaluation process simultaneous with the inference engine under the PC Plus development environment. PC Plus first identifies the rules which are concluding the root-frame's goal parameter. Until the goal parameter is completely established, the identified rules are individually evaluated in a sequence in which the inference engine locates them. The developer reviews the sequence list and makes necessary corrections by assigning UTILITY numbers to the rules to control the actual order in which the rules are evaluated. Setting the correct order of execution enables pruning

of the rules by speeding up the inference engine's rule evaluation process. Moreover, the correct order of execution eliminates the unnecessary questions asked by the inference engine.

During the evaluation process, the developer can ask "why" a rule is tested to understand the link between the rules within the knowledge base. When the "why" question is asked, PC Plus displays the rule's premise and conclusion as well as the reason of testing that particular rule. The "why" question can also be used for determining why a question is asked during a consultation session. Any error in writing the rules can easily be identified this way and can be corrected accordingly.

Value assignment to parameters can be verified through the "how" question tool. When a "how" question is asked about a parameter, PC Plus replies showing how the parameter received that particular value by displaying the rule which determined the value for that parameter in question. In addition, the "how" question can be used for testing individual assignment of values to the parameters, and can also be used to test the data imported from the input text files.

A review tool is a useful command to test the knowledge base under different responses to the questions asked by the inference engine. The developer can change the responses to the questions which are listed by the review tool for testing purposes. A different response results in a different outcome of the consultation. By changing the responses to the questions,

the developer can verify the rules that would not have been used in the initial response.

Once the knowledge base rules are determined to be error-free, the initial testing procedure continues by entering hypothetical conditions and evaluating future predictions given by ASPX. In this stage, the developer can identify rules that lead to incorrect suggestions which can then be altered to suit the needs of the selected refinery plant operations. The initial testing is concluded when no rules are detected to have problems, and no more incorrect suggestions are given by ASPX under comprehensively tested conditions.

3.2.2 Implementation and Intermediate Testing

Prior to the intermediate testing procedure, implementation of ASPX in the selected plant is necessary to obtain a real-time testing of ASPX. The implementation procedure consists of setting up a computer in the selected plant and giving a brief course to the plant operators who are going to use the ASPX program. An 386-20 MHz PC, which is capable of running ASPX fast and efficiently, is installed in the selected plant. The installation of the program consists of copying necessary files to the hard disk and assigning permits to operators to use ASPX. Because the ASPX program is not completely finished, only designated operators are allowed to use the program in a secluded room to avoid premature exposure of an unfinished computer program to the rest of the crew. Each of these

designated operators has a login name and a password which enable them to access the program. After assigning permits, the ASPX program is ready to be tested further.

The objective of the intermediate testing is to customize the ASPX knowledge base for the selected plant real-time operations. The testing procedure is performed by designated plant operators who know the selected plant operations and the plant's performance. These designated operators use the program on an everyday basis by entering values to the data base and by running the expert system shell to test the ASPX program's performance. Any incorrect suggestion given by ASPX is recorded in the log by the operator who encountered the problem, and is fixed later by the developer based on the origin of the problem. Possible modifications include adding to, or changing the rules in the knowledge base, and such fundamentals as changing the logic trees. The intermediate testing will continue until all necessary adjustments are made in the knowledge base and in the user-interface part, and all the designated operators are satisfied with the performance of the ASPX program.

3.2.3 Advanced Testing

The fine tuning of the ASPX is achieved by the advanced testing procedure. In this final testing phase, the entire personnel are allowed to access ASPX and to apply it to the real-time operations under supervised conditions. Although the operators will always be supervising the

computer's suggestions, the computer will have a significant effect on the process control strategy. In case of any unacceptable suggestion given by the computer, experienced operators will be consulted to locate the source of the malfunction. The advanced testing will reveal required adjustments to be made based on the feed-back received from the crew and the supervisors. Further adjustments to the program will be made to improve the accuracy of the ASPX program until no more improper suggestions are given by the ASPX program.

The final ASPX product is achieved upon completion of the advanced testing procedure. The final ASPX will perform equal to an expert operator who has at least 20 years of practical experience in the selected refinery plant.

4 Application of ASPX to the Selected Refinery Activated Sludge Plant

The selected refinery activated sludge plant is suffering from a frequent loss of nitrification capabilities due to the plant's inadequate process control techniques, limited understanding of unit processes' behavior and their capabilities, and misjudgment made in process control decisions by inexperienced operators. Moreover, frequent personnel rotation within the plant compounds the problems.

Refinery process-water contains high levels of toxic chemicals, high COD, considerable amounts of phenol, high concentration of sulfides, variable pH range, and high oil and grease content. Property of refinery waste makes the treatment of refinery process-water important because violation of NPDES standards cause significant damage to the environment.

Refineries are under the constant scrutiny of the public because the public is well aware of consequences of refinery spills to the environment. This public awareness forces the refinery management to take special actions in the process-water treatment plants to avoid confrontation with the public and the closely monitoring government agencies.

One of the state-of-the-art solutions is the use of expert system technology with the decision making process in controlling the process-water treatment plant operations. ASPX is a knowledge-based expert system which is being adapted to the selected refinery plant as a new process control

option. Because ASPX is developed exclusively for nitrification problems within the selected refinery plant, it is our belief that ASPX will significantly overcome the problems which the plant is experiencing currently.

ASPX's basic task is to detect nitrification upsets and give suggestions on how to solve these nitrification problems. Development of ASPX involves building the foundation of ASPX, and testing procedures which confirm the integrity of the expert system. Building the foundation of ASPX is explained in section 3.1 including the knowledge acquisition techniques (section 3.1.1), structural components of ASPX (section 3.1.2), knowledge base construction (section 3.1.3), development of graphical user-interface (section 3.1.4), data base functions (section 3.1.5), statistical and mathematical models, PC Plus expert system shell (section 3.1.6), and the development of report generation functions (section 3.1.7).

At the developing stage, ASPX is performing at the level which a new operator would perform the first day on the job. However, through continuous development and exhaustive testing procedures, the ASPX final product will provide advice at the level of an operator who has 20-years of practical experience in the selected refinery plant.

Upon completion of advanced testing, ASPX finds a number of different applications in the selected plant including establishing a unique process control technique, regulating the record keeping process, and monitoring the report generation process. Furthermore, additional possibilities of

applications are feasible such as performing statistical and pattern analysis on the data base records, or using ASPX as a training tool.

The primary application of ASPX in the selected refinery plant is to establish a state-of-the-art process control technique which will resolve the nitrification problems within the plant. ASPX is capable of controlling the plant operations successfully by reducing problems caused by nitrification failure without extensive cost and effort requirements. The control of the plant processes by ASPX is achieved by monitoring process control parameters, and making proper adjustments to the process control equipment according to the recommendations given at the end of running the ASPX expert system. Detailed analysis of controlling the selected plant by ASPX is given later in the text (see section 4.3).

The second most important application of ASPX is the ability to maintain a regular record keeping process. Record keeping is not only required by law but it is also an essential part of operating a treatment plant since the history of the data base has a significant effect on the decision making process. Record keeping can be regulated by ASPX in a manner that each operator is monitored during the data base record keeping process. Because ASPX keeps track of the operator's actions, the supervisor can review the operator activity logs and can identify operators who do not comply with the record keeping regulations. The possibility of falsifying records is also eliminated since only the supervisors have authority to alter the previous data base records. Full functions of the ASPX data base component is given in section 3.1.5.

ASPX provides another application field, the report filing process, which is a regular practice in treatment plant operations. Compliance with the report filing requirements is a problem in almost all wastewater treatment plants because of the extended size and the frequency of reports required by the management. ASPX report generation functions helps the operators to comply with these regulations by making the report generation process easy. ASPX can generate hourly, daily, or weekly reports by requiring no additional input but the data base records. The abnormalities, suggestions, and actions performed during the time the report is covering are included in the report. Graphical presentation and pattern analysis of data base parameters are also available to the plant operators for performance review purposes. Because report generation is made easy by the ASPX program, the operators will less likely oppose the record filing requirements. A complete description of the report that can be generated by ASPX is given in section 3.1.7.

Additional possibilities of ASPX application in the selected plant includes obtaining statistical and graphical representation of data base records which can be used in reviewing the plant's overall performance. The information obtained from ASPX is an invaluable tool for reviewing the past performances of the plant since the data presented in a graphical format provides a more feasible connection within the data base parameters. Parameter tracking and variable interrelation graphs enable operators to understand the plant's unit processes to a degree never before possible.

The possibilities of using ASPX as a trainer is a potential aspect for future developments of the program. Currently, ASPX is capable of teaching "practical experience" within a short time period and without risking unsafe real-time trial and error attempts with the plant's processes.

However, a more sophisticated training component needs to be developed before operators can take full advantage of the expert system's teaching ability. The development of the ASPX training component will provide a well-suited teaching environment for any operator in the plant, especially the new inexperienced operators.

4.1 Description of the Selected Plant

A description of the selected refinery activated sludge plant is provided in this section to reveal the process control parameters and the equipment that can be used to control the plant operations. The selected refinery activated sludge plant treats the process-water which is discharged from its oil refinery complex. The treatment plant consists of oil separators, feed tanks, a polymer injection and a mixing chamber, dissolved air floatation (DAF) tanks, phosphoric acid injection tanks, an equalization basin, a caustic addition tank, aeration basins, clarifiers, and an aerobic sludge digester. The selected plant's process diagram is shown in Figure 4-1.

The oily process-water is first treated with four American Petroleum Institute (API) gravity-type oil-water separator tanks. The API separators

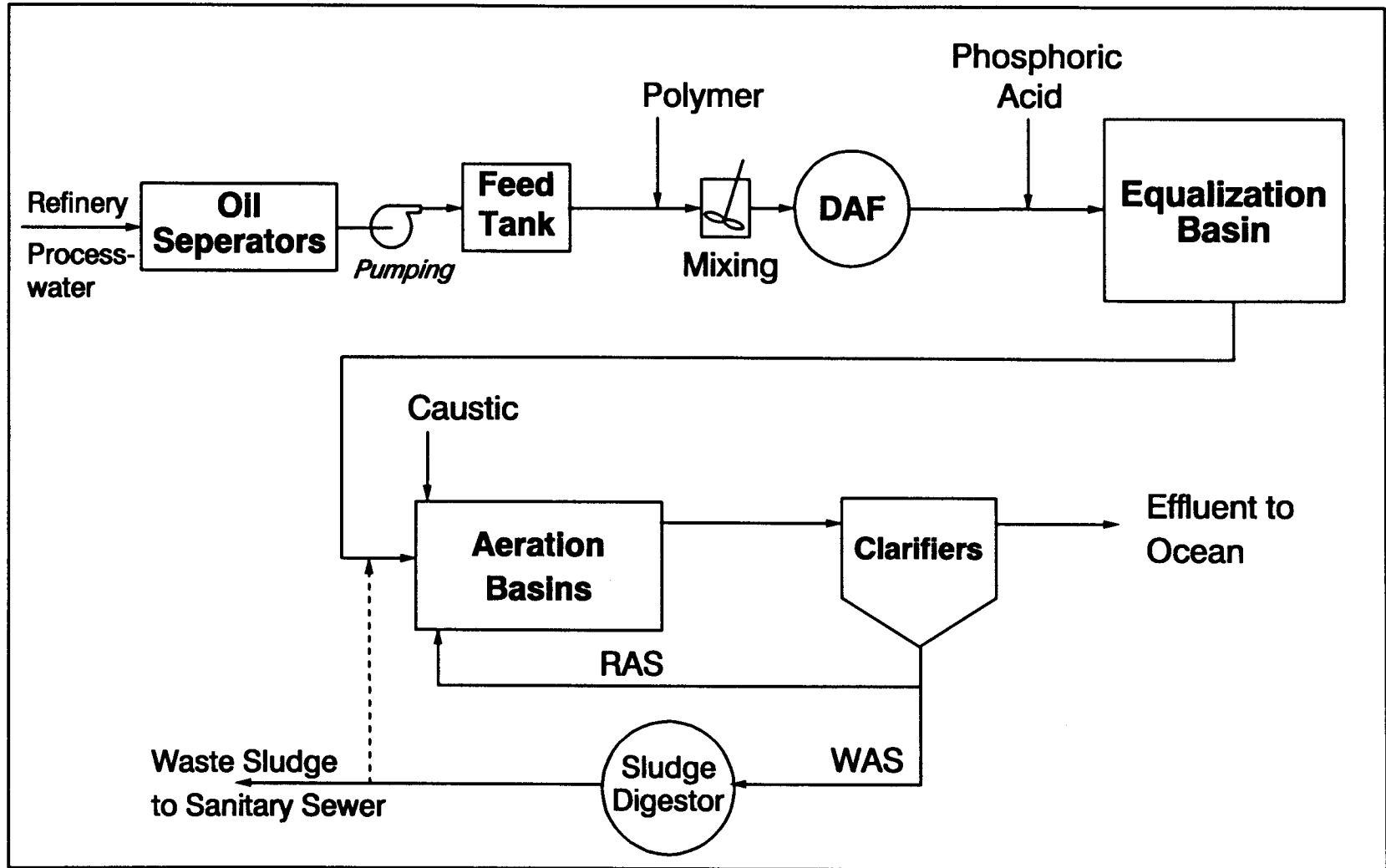


Figure 4-1: The process diagram of the selected refinery treatment plant

are commonly used in refineries because they successfully capture 60-99% of the oil before it is discharged to the treatment plant. The API separator is a long, large volume tank where the oily process-water enters at one end and moves slowly towards the other end while allowing the oil to separate from the water phase. The oil is recovered by skimming the floating oil from the surface at the effluent end of the tank. The recovered oil is then sent back to the refinery complex to be treated. The efficiency of the API separator is greatly influenced by the flowrate, the suspended solids concentration, size of the oil bubbles, and the temperature of the oily process-water.

After the separation, the water phase is pumped to the treatment plant from the bottom of the separator tanks to the feeding tanks which are located at a higher elevation than the rest of the plant process to allow flow through the plant by gravity. There are two feeding tanks where the process-water is kept for a short retention period before it is released to flow to the DAF units.

A typical DAF unit is a covered, deep, narrow tank where high pressure air is introduced in fine bubble forms at the bottom of the tank to coagulate and to float solids to the surface. Before the DAF units, polymer is injected into the process-water to amplify the degree of coagulation. After polymer is introduced, the process-water flows through a rapid mixing chamber where the water is vigorously agitated for a very short time to activate the full effects of the polymer coagulation process. The process-water then flows into flocculation chambers where the water

is stirred slowly to establish a contact between the polymer and the small particulate to form a larger floc.

After a short residence time, the coagulated process-water enters the floatation tanks where the water, highly saturated with air, is released from the bottom of the tank. The saturation process takes place in air saturation vessels where compressed air is introduced to the recycled stream taken from the effluent of the DAF units. The water in the air saturation vessels is highly saturated with air before it is emitted to the floatation tanks. When the saturated water is released to atmospheric pressure at the bottom of the DAF tanks, because of the substantial change in pressure, the air forms microscopic bubbles which attach to the large flocs of coagulated particles by lowering the particles' specific gravity. Low specific gravity causes the solid particles which are bonded with air bubbles to float to the surface of the DAF tanks. Solids are finally skimmed from the top of the DAF tanks and taken to the sludge handling facilities.

After the DAF units, phosphoric acid is injected into the process-water to overcome problems caused by the nutrient deficiency. Included among the reasons for choosing phosphoric acid as a phosphate source are ease of handling, less cost, and flexibility in controlling the injection rate.

Subsequently, the process-water flows into a covered equalization tank where the water is aerated for mixing purposes and for avoiding anaerobic conditions. Although the primary objective of the equalization basin is to regulate the flowrate going into the aeration basins, it is sometimes used

as a holding tank when a shock loading is received such as high ammonia, or high organic loading. The process-water flows into two channels at the end of the equalization tank where it is directed into the aeration basins.

At the entrance of each activated sludge unit, caustic solution is injected into the tanks to increase the pH to around 6.8 - 7.4 for the optimal growth of the microorganisms. Separate mixing chambers for caustic addition are eliminated because the chemical is introduced to the aeration tanks where the water is actively mixed. Caustic is added through pipelines connected to a holding tank which is adjacent to the aeration basins. A mechanical level indicator is installed on the caustic tank to display the caustic level in the tank and to show the rate of caustic addition.

There are two symmetrical activated sludge units which are coupled with one sedimentation tank for each unit. Activated sludge units are conventional, diffused air aeration units with the capability of returning sludge to the beginning of the aeration tanks. Air is introduced by fine bubble diffusers which are located at the bottom of each aeration tank. There are two 350 HP and one 450 HP air blowers on the premises with one being a standby. The air blowers are single-speed pumps where the air flow can be controlled by throttling the blower discharge valve. The blowers are equipped with several pressure gauges on both the discharge and the suction side of the pipelines where air filters are placed for infiltrating the intake air at the suction side.

There is one sedimentation tank at the end of each aeration basin, with a total of two in the plant. The sedimentation tanks are circular with typical notch weirs and a sludge collection system. The sludge is collected at the bottom-center of the sedimentation tanks to be wasted and to be recycled. The portion of this sludge is returned back to the aeration basins to keep adequate amounts of bacteria in contact with the wastewater. Excess sludge is pumped to the aerobic sludge digester where the sludge is stabilized until it is disposed of in the sanitary sewer.

Although the primary purpose of the sludge digester is sludge stabilization by endogenous consumption of organic compounds, it also serves as a backup bacteria storage place. Use of the aerobic digester is exceptionally beneficial to the selected refinery treatment plant because when a toxic shock destroys the living cells in the aeration tanks, the aerobic digester is used as a source of new bacteria cells. This way, the supplementary bacteria mass can be supplied within a very short time and the plant can quickly recover from the effects of a toxic shock.

Effluent water from the sedimentation tanks is mixed with the effluent of the clean water system which treats the storm drainage water, cooling tower and boiler blowdown. The clean water system consists of API oil-water separators, Induced Air Flootation (IAF) units, and concrete mixing chambers where two effluents of waste and clean water treatment merge before it is discharged to the ocean. Mixing clean water with the process-water treatment plant effluent dilutes the concentration of the chemicals in

the water. Because of this dilution, the risk of violating the NPDES permit is significantly reduced.

4.2 Process Control of the Plant Prior to ASPX

The selected refinery activated sludge plant is equipped with a combination of electrical and mechanical devices to monitor the plant. Some of the monitoring equipment includes the following devices: flow meters, DO probes, pH meters, air pressure gauges, chemical level indicators, sludge depth indicators, and temperature probes. Besides monitoring, laboratory analysis of several parameters is required including the TOC test, sulfide concentration determination, ammonia-nitrite-nitrate testing, phenol concentration measuring, total suspended solids (MLSS) determination, and the SVI test.

For individual unit processes, process control equipment and parameters are needed to establish an understanding of what to control and how to control that process. For DAF units, the process control parameters include flowrate, TSS, and the oil content of the process-water. The process control parameters indicate what to control in a unit process. The process control equipment is used in DAF units to control these parameters by adjusting the polymer injection rate, and by controlling the compressed air pressure.

The equalization basin consists of two pieces of process control equipment which are maximum/minimum level controllers, and flow deviation equipment. Based on the incoming flowrate, the level controller releases the water flow accordingly. In case of maintenance, the flow may be only directed to one activated sludge unit by the flow deviation equipment.

Activated sludge units are more complicated than the rest of the processes in the plant because the operations are not only dependent on liquid handling but also aeration equipment handling. A number of controls exist on air blowers such as ON/OFF switches, the suction pressure control valve, and the discharge air flow control valve. Since the blowers run at a constant speed, the amount of air can only be controlled by throttling either suction or discharge air valves. The DO probes which are installed in the aeration basins continuously measure the DO concentration and display it in the nearby display case where the gauges are located.

Furthermore, pH adjustment equipment needs to be controlled to assure feasible living conditions for the bacteria. The pH adjusting equipment controls include a level indicator on the caustic tank and a metering pump to release the desired amount of caustic into the aeration basins. The pH electrodes read pH continuously in the aeration basins and display it on gauges where the DO is displayed.

The clarifiers are equipped with sludge collection systems, return activated sludge pump controls, and a waste activated sludge pump ON/OFF switch. Based on the conditions existing in the aeration basins, the equipment on

the clarifiers are adjusted accordingly. The clarifiers are also equipped with sludge blanket depth measurement gear which is mostly used to indicate the time for wasting sludge.

The control room located in the plant consists of several readings received from the field measurements. Operators observe the condition of the plant from the control room by the gauges which are connected to outside measuring equipment. Gauges like the DO readings in the aeration basins, RAS pump speed, pump ON/OFF indicator light, and sludge levels in clarifiers, all help operators to observe the effluent quality directly from the control panel. The control panel has the ability to control some equipment directly without leaving the control room (*e.g.*, pump ON/OFF switch). Other equipment requires the operator's attention at the location, for example, adjusting the air suction valve.

Currently, the most common type of process control in the selected refinery activated sludge plant is executed on field location while the operator makes routine observations of the plant. If the operator observes any abnormalities such as excessive aeration, he/she checks the controlling equipment to ensure the proper operations. Some monitoring is made in the control room. When the operator catches a faulty reading on the control panel, necessary actions are taken.

The process control technique for DO is performed as follows: operator detects a low DO reading in one of the aeration basins. The operator then makes an adjustment to the blowers based on his/her experience. The

operator observes the DO change resulting from the new adjustment of the blowers until a steady state DO value is read. If the new DO reading is satisfactory, the operations continue until a problem is detected.

However, more complex situations may exist if the adjustment made on the blowers does not provide a satisfactory DO concentration. The operator, in this case, readjusts the blower valves by the trial and error method until a steady state of desired DO level is achieved. This procedure may take several adjustments and considerable time. During this trial and error period, the aeration process is most susceptible to problems because the DO value fluctuates rapidly causing unsteady state responses from the activated sludge process.

The process control of pH adjustment prior to ASPX also was based on the trial and error method where the operator releases a certain amount of caustic to the aeration basins and records the pH reading after a little while. If the pH reading is not acceptable, more or less caustic is released based on the pH reading. This adjustment takes place continuously since the influent characteristics of the process-water changes constantly.

Similarly, phosphoric acid control is executed by trial and error where the effluent phosphate concentration is observed and the excess or shortage amount is estimated accordingly. When a shortage of phosphate exists, the activated sludge process performance is greatly affected and nitrification will likely be lost. Since the problem is not detected until it actually

happens, phosphoric acid control must be improved to eliminate the problems with the nitrification process.

The return activated sludge control is accomplished based on the incoming flowrate and the condition of the activated sludge process. The operator adjusts the RAS pumps accordingly to meet 75-100% return rate of the influent flowrate.

Waste activated sludge control is initiated based on the sludge blanket depth in the clarifiers. Prior to ASPX, the plant was not operated under sludge age which is the primary parameter to control the waste sludge rate. Since the refinery plant experiences frequent toxic shocks where the entire bacteria mass die, wasting sludge is very rare. Because the growth rate is very low when an inhibitor compound is received by the plant, the sludge is conserved by not wasting at all. On rare occasions when the sludge blanket reaches a level in which the clarifier no longer can hold the sludge volume, the collected sludge is wasted to the sludge digesters.

The most common problem observed in the selected plant is the nitrification failure. Inadequate process control techniques and the nature of the process-water contributes to the frequent failure of the nitrification process. The nitrification process is greatly affected by the dissolved oxygen concentration, sludge age, pH reading, temperature, phosphate concentration and the incoming ammonia concentration. Since all these parameters are interrelated, and there is no clear cut understanding of the

individual impact on the nitrification process, the trial and error process control method is an inadequate exercise of process control methodology.

A better process control strategy for the selected refinery activated sludge plant is essential to ensure effective treatment of the process-water. Since the process control equipment and the plant layout itself are fairly simple in the plant, application of expert system technology is also an equally simple exercise for this plant. ASPX can benefit the plant enormously by eliminating the potential problems with the nitrification process and by regulating the process control technique which will not only correct the problems but also prevent them from happening. Results achieved after the application of ASPX to the selected refinery activated sludge plant will show significant improvements on the effluent quality and the operational values of the plant.

4.3 Plant Process Control by ASPX

ASPX provides two invaluable process control options. First, the detection and the solution to the nitrification problems occurring now, and, second, the prevention of future problems before they take place.

ASPX is an efficient preventive process control tool since the ASPX expert system can identify potential problems before they actually occur and before they cause any damage. ASPX can detect potentially dangerous stages by analyzing the data base, the knowledge base, and the

mathematical and statistical models. Predictions are made based on monitoring the plant variables such as the influent process-water characteristics, and an increase in ammonia loading. The statistical model provides useful information of the likelihood of problems occurring in the future. The mathematical model can provide additional information about the process in numerical values. Once the potentially dangerous stages are identified, ASPX suggests preventive process control measures.

Although prevention of problems is always beneficial, the primary design objective of ASPX is to detect and to correct the problems which are occurring at the present time. ASPX is exclusively developed for the control of activated sludge units in the selected plant to eliminate the problems caused by the nitrification failure. Therefore, the remainder of this document covers the process control parameters of activated sludge units and the clarifiers.

The activated sludge unit parameters include the dissolved oxygen concentration, sludge age, caustic addition rate, phosphate concentration, organic loading, ammonia load, MLSS, temperature, and flowrate. The parameters considered in the clarifiers are the return sludge rate, waste sludge rate, sludge blanket depth, and sludge MLSS. Table 4-1 gives the complete list of process control parameters, process control equipment, number of locations which the parameter is measured, and monitoring equipment which ASPX uses during the process control of the selected plant.

Table 4-1: The list of process control parameters, process control equipment, location and the monitoring equipment for the ASPX process control technique

Unit Location	Process Control		Monitoring
	Parameter	Equipment	
Activated sludge	Dissolved oxygen	Blower ON/OFF switch	DO probe
		Blower discharge valve	
	Sludge age	Sludge waste pump	Calculated, Eq. 2-2
	pH reading	Caustic tank level indicator and metering pump	pH electrode
	Phosphate concentration	Phosphoric acid tank level indicator and metering pump	Lab analysis of effluent phosphate concentration
	TOC (organic loading)	Temporary storage in holding tank	TOC analyzer
	Ammonia load	Temporary storage in holding tank	Lab analysis of influent and effluent ammonia-nitrite-nitrate concentrations
	MLSS	Sludge waste rate	Laboratory MLSS analysis
		Return activated sludge rate	
	Flowrate	Temporary storage in holding tank	Flow meters
Temperature	Temporary storage in holding tank	Temperature probes	
Clarifiers	Return activated sludge rate	RAS pump controls	RAS flow meter
	Waste activated sludge rate	WAS pump controls	WAS flow meter
	Sludge blanket depth	RAS/WAS pump controls	Sludge level indicator
	Sludge MLSS	No apparent control	Laboratory MLSS analysis

Process control of the plant by ASPX consists of monitoring the process control parameters, adjusting the corresponding equipment settings based on the advice given by the ASPX expert system consultation. The process control scheme is illustrated in Figure 4-2 and listed in the following steps:

- 1) Monitor the process control parameters listed in Table 4-1 by the monitoring method listed in the same table.
- 2) Enter the values monitored into the data base using the ASPX user-interface program where the data base functions can be executed.
- 3) Run the ASPX expert system to diagnose abnormalities with the nitrification process and to receive advice on how to solve the problems.
- 4) Review the results generated by ASPX after the expert system consultation on solving the nitrification problems.
- 5) Make recommended adjustments on the process control equipment to correct the current problem, or to prevent future problems.
- 6) Confirm adjustments after ½-1 hour depending on the type of adjustment made to observe the progress of the plant condition.
- 7) Go to step 1 and exercise the use of ASPX process control procedure regularly.

The monitoring process which determines the status of the plant is implemented three times a day every 8 hours, on a regular basis. Although, currently, twenty-one parameters are monitored in the plant, six of the twenty-one parameters influence the process control decisions the most. These six parameters are as follows: sludge age, dissolved oxygen, pH, ammonia concentration, temperature, and phosphate concentration.

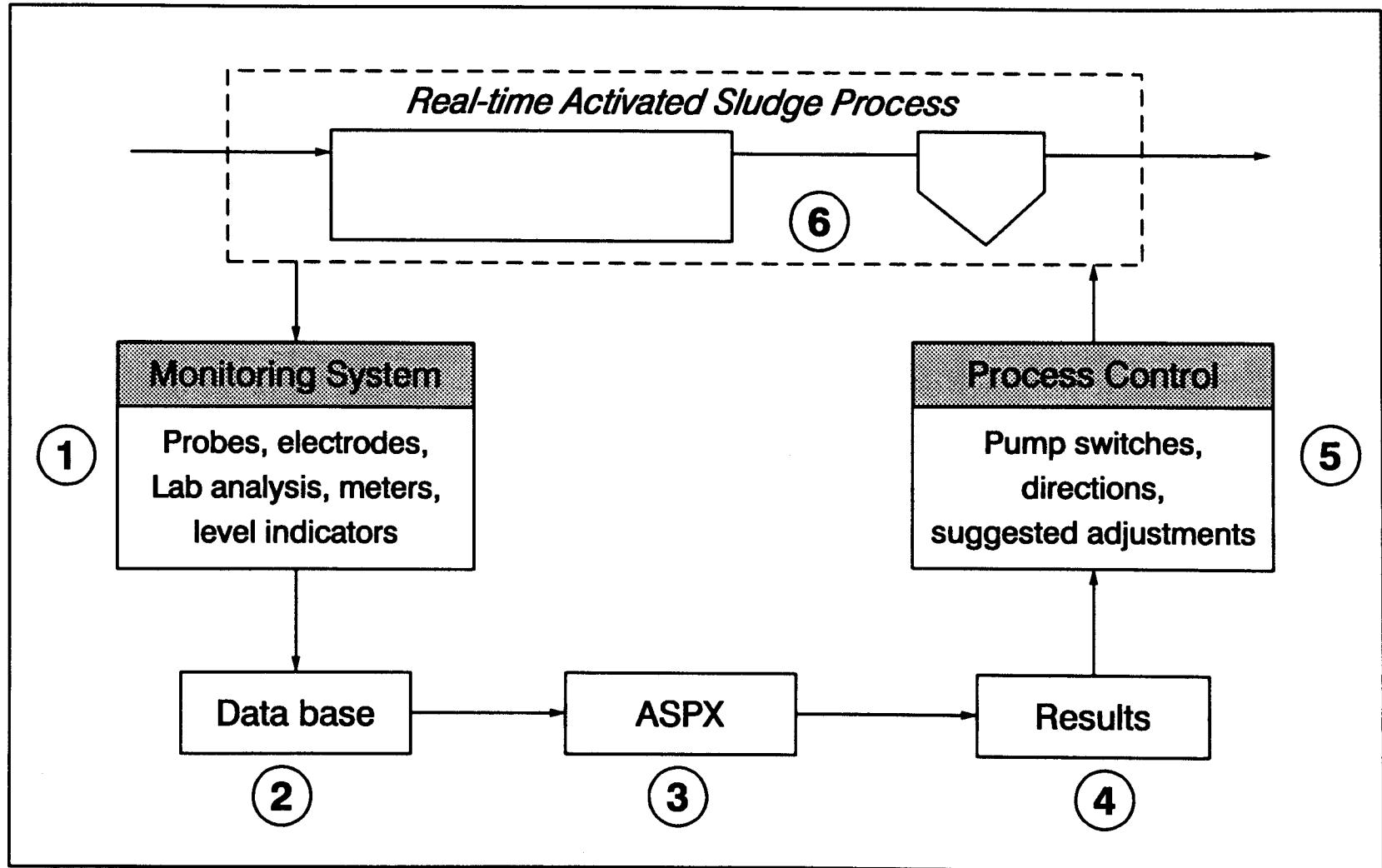


Figure 4-2: The process control scheme of the ASPX program

All six parameters with the exception of sludge age is measured using probes, electrodes, and laboratory techniques. The sludge age is calculated indirectly by the waste sludge rate, waste and activated sludge MLSS concentrations, flowrate, and the effluent TSS concentration (see Eq. 2-2).

The values of the process control parameters are entered into the ASPX data base through the user-interface program for record keeping and process evaluation purposes. The data is entered three times daily under the internal checking process of the interface program which detects errors in recording, as well as declaring the missing parameters in the data base records. The values in the data base are used to establish the current state of the nitrification process in the plant.

Once the values are entered into the data base, ASPX runs on an 386-20 Mhz computer in the control room, separately from any other process in the plant to diagnose problems with the nitrification process. During the consultation session, the knowledge base is accessed and current parameter values are loaded into the knowledge base rules to evaluate the nitrification problems. The expert system also accesses the statistical and mathematical models to receive additional information on the plant's problems.

At the end of the consultation session, ASPX displays the problems which are detected by the expert system. The result may suggest an adjustment to the process control equipment, or may direct operators to take a specific action against future problems. Unlike the expert person's heuristic

knowledge, ASPX is capable of explaining "why" and "how" a decision is made. The program displays a list of reasons that are used during the expert system's inference procedure. Each reason carries a "because" clause to explain the purpose of the reason. Declaring the reasoning procedure gives the operators additional confidence about suggestions given by the ASPX program. If the program worked like a "black box," the operator would have had doubts about the advice because he/she would not know how the computer reached this specific conclusion.

The explanation utility is also useful to detect problems within the reasoning procedure. If any problem is detected with the ASPX reasoning procedure, the operator disregards the suggestion given by the program and notifies the supervisor about possible error in the reasoning procedure. Errors in the program can be corrected with the procedures outlined in section 3.2 for testing the ASPX program.

After reviewing the results given by ASPX, the operator takes corrective actions which are suggested by the ASPX expert system. The suggested actions vary depending on the nitrification state which is identified at the beginning of the expert system consultation. The most common actions suggested by ASPX are making adjustments to the process control equipment, and testing the integrity of the monitoring equipment.

After an adjustment is made, there is a certain lag time until the effects show. This lag time widely depends on the type of adjustments made, and type of equipment in question. For example, an increase in the caustic

addition rate has relatively a much faster response than an adjustment made on the waste activated sludge pumps. After an action is taken, ASPX advises a certain lag time before confirmation of the adjustment is made.

If the above mentioned process control steps are regularly practiced, ASPX may greatly benefit the plant operations by reducing the risk of violating the NPDES permit and by regulating the steady state plant operations.

In the following sections, ASPX process control technique for individual unit processes is explained in detail including the process control parameters, process control equipment, and the monitoring systems. The individual process control techniques are illustrated in the form of logic trees. The legend for the logic trees given in the following sections are illustrated in Figure 4-3. The complete list of the knowledge base rules created in this dissertation to control the nitrification process may be found in Appendix E.

4.3.1 Ammonia Control by ASPX

Ammonia control is an essential part of the nitrification process control routine because the nitrification process can be described in terms of ammonia (NH_4^+), nitrite (NO_2^-), and nitrate (NO_3^-) concentrations. The nitrogen concentrations are used to calculate the ammonia reduction

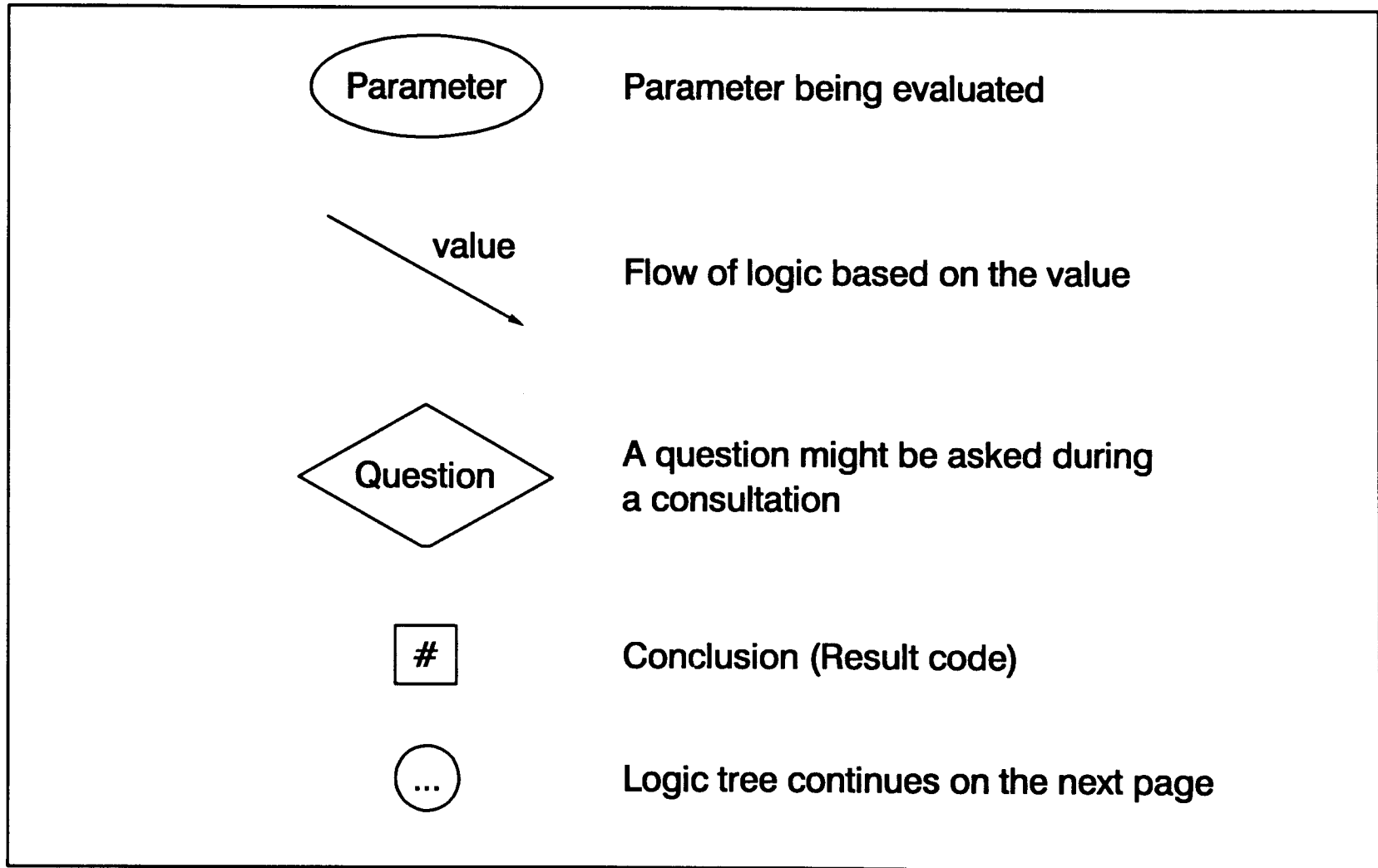


Figure 4-3: Legends used for the logic trees

efficiency achieved during the nitrification process. The ammonia reduction efficiency is measured by the nitrification rate calculation which indicates the amount of ammonia consumed by the amount of bacteria mass per day.

The nitrification problems can be identified by monitoring MLVSS, influent and effluent ammonia, and nitrate concentrations. If ammonia concentration exceeds the maximum allowed limit in the effluent, or the calculated nitrification rate is below the desired value, the nitrification process is said to be failed.

Process control parameters involving ammonia control are shown in Figure 4-4 which is the logic tree behind the process control technique for ammonia concentration. The process control parameters included in the ammonia control are as follows: the effluent ammonia-nitrite-nitrate concentration, the nitrification rate, and the influent ammonia concentration. Moreover, the question of reducing the influent flow is asked of the operator to differentiate the answers.

The effluent ammonia-nitrite-nitrate concentrations, which are used to detect problems with the nitrification process, are monitored under daily routine laboratory analysis. The effluent concentrations of nitrogen are the principal indicators of effluent water quality which must meet the NPDES permit.

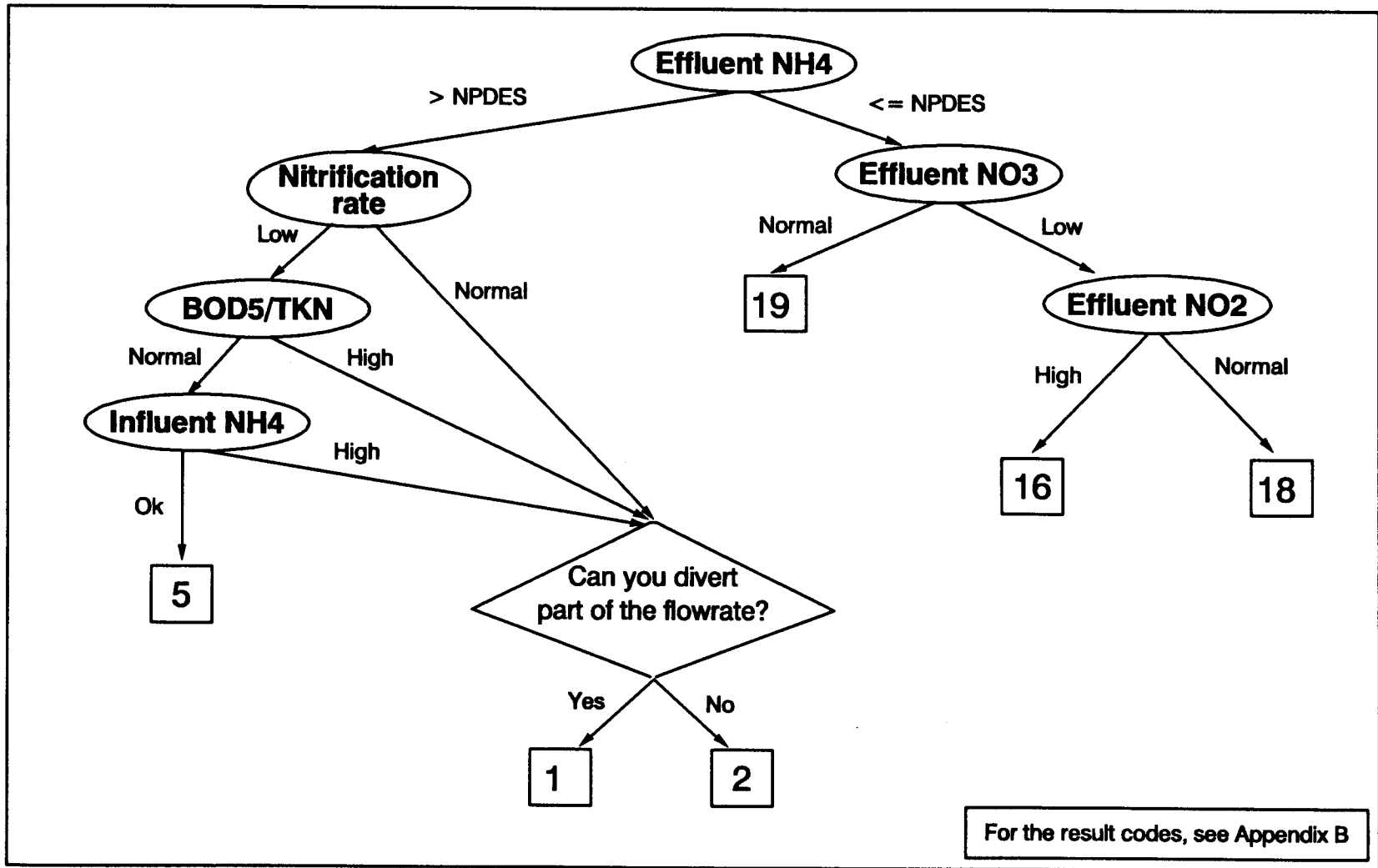


Figure 4-4: The effluent ammonia concentration process control logic tree

The efficiency of the nitrification process is indicated by the nitrification rate and it shows the success in reduction of the ammonia concentration in the plant. The nitrification rate calculation is used to evaluate the overall plant performance in achieving the nitrification state. The rate is calculated by using the influent and effluent ammonia concentrations as shown in the following equation:

$$NR = \frac{(NH_{4,in}^+ - NH_{4,out}^+) (Q + Q_r) (24) (60)}{V X_{vss}} \quad (4-1)$$

where

- NR = Nitrification rate, mg NH₄⁺-N/mg MLVSS · d
- NH_{4,in}⁺ = Ammonia concentration in the influent, mg/L
- NH_{4,out}⁺ = Ammonia concentration in the effluent, mg/L
- Q = Flowrate, gpm
- Q_r = Return activated sludge flowrate, gpm
- V = Volume of aeration tank, gal
- X_{vss} = MLVSS concentration, mg/L

Influent ammonia concentration alone has an additional impact on the process control of nitrification since the ammonia load is a key parameter in detecting the potential nitrification problems. The detection of problems is performed by monitoring the influent ammonia concentration on an hourly basis, three times a day. If the ammonia load exceeds the plant's design limitations, there is strong evidence that the nitrification process would soon fail.

The equipment involved in the process control of ammonia is limited due to inability to manipulate the concentration of ammonia coming into the plant. When the plant receives a high ammonia load, one of the few actions that can be taken includes the reduction of the flowrate by holding the incoming water in the equalization basin. Although the control of flowrate can be performed by throttling the effluent valves of the equalization basin, reducing the load is not always feasible because of the limited storage availability in the basin at the time of the request.

In addition to high ammonia loading, there are a number of other reasons for the failure of the nitrification process. Some examples are the existence of improper conditions in sludge age, dissolved oxygen, pH, and phosphate concentration. These parameters are individually investigated in the following sections.

4.3.2 Sludge Age Control by ASPX

Sludge age is one of the most important process control parameters that affects the nitrification process. The sludge age reflects the length of time which the bacteria mass resides in the aeration basins and it is calculated based on the flowrate, wasting rate, and solids in the system (see Eq. 2-2). The control of all the parameters which involve sludge age calculation, excluding the sludge wasting rate, is difficult. Thus, the wasting rate is used as a primary control of sludge age based on reduction in the sludge waste rate which would increase sludge age.

Because the nitrifying bacteria, *Nitrosomonas* and *Nitrobacter*, has a slow growth rate, high sludge age (usually more than 20 days) is required to achieve complete nitrification. Sludge age is a powerful indicator of whether the nitrification would take place or not. Sludge age is widely used in the wastewater treatment plants as a primary process control parameter.

The logic tree for the process control scheme of sludge age is shown in Figure 4-5(a) and 4-5(b). The process parameters which influence the sludge age control include MLSS, waste sludge MLSS, sludge blanket depth, sludge wasting rate, return sludge rate, OUR test, and F/M ratio.

The MLSS reading plays a significant role because it is an indicator of the amount of bacteria mass present in the aeration basins. The MLSS concentration is monitored daily by laboratory analysis of suspended solids. If the MLSS reading registers below allowed limits, the nitrification would likely fail because there is not enough bacteria mass to consume the incoming ammonia load. Although the MLSS reading represents the total bacteria mass, the amount of nitrifier bacteria is closely correlated with the amount of carbon consuming bacteria, both of which can be estimated based on single MLSS measurement. The MLSS can be controlled by both the return activated sludge and the sludge wasting rate.

The sludge MLSS concentration is monitored daily by laboratory analysis and it is used as a process performance parameter. Although the sludge

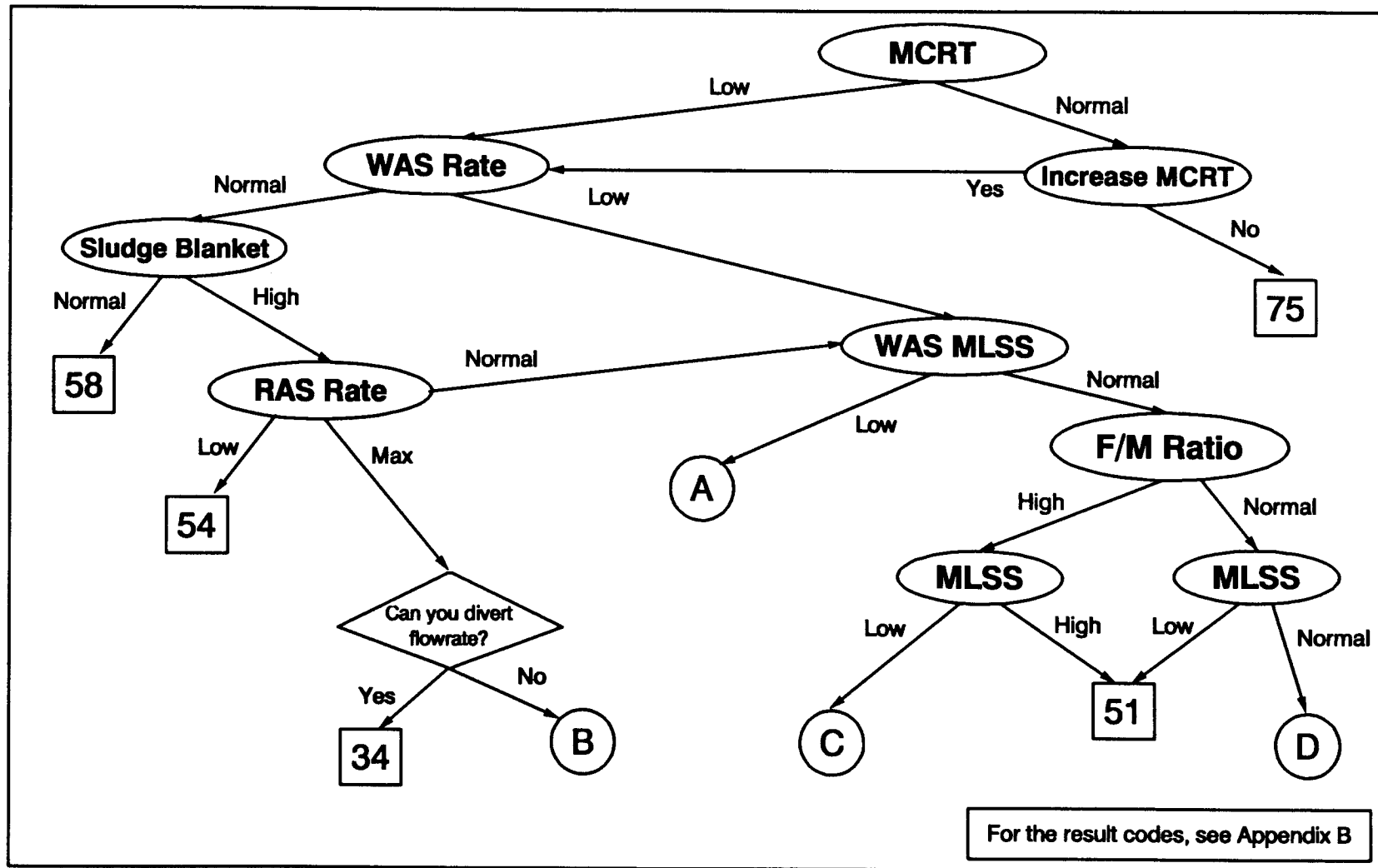


Figure 4-5(a): The sludge age process control logic tree, part 1

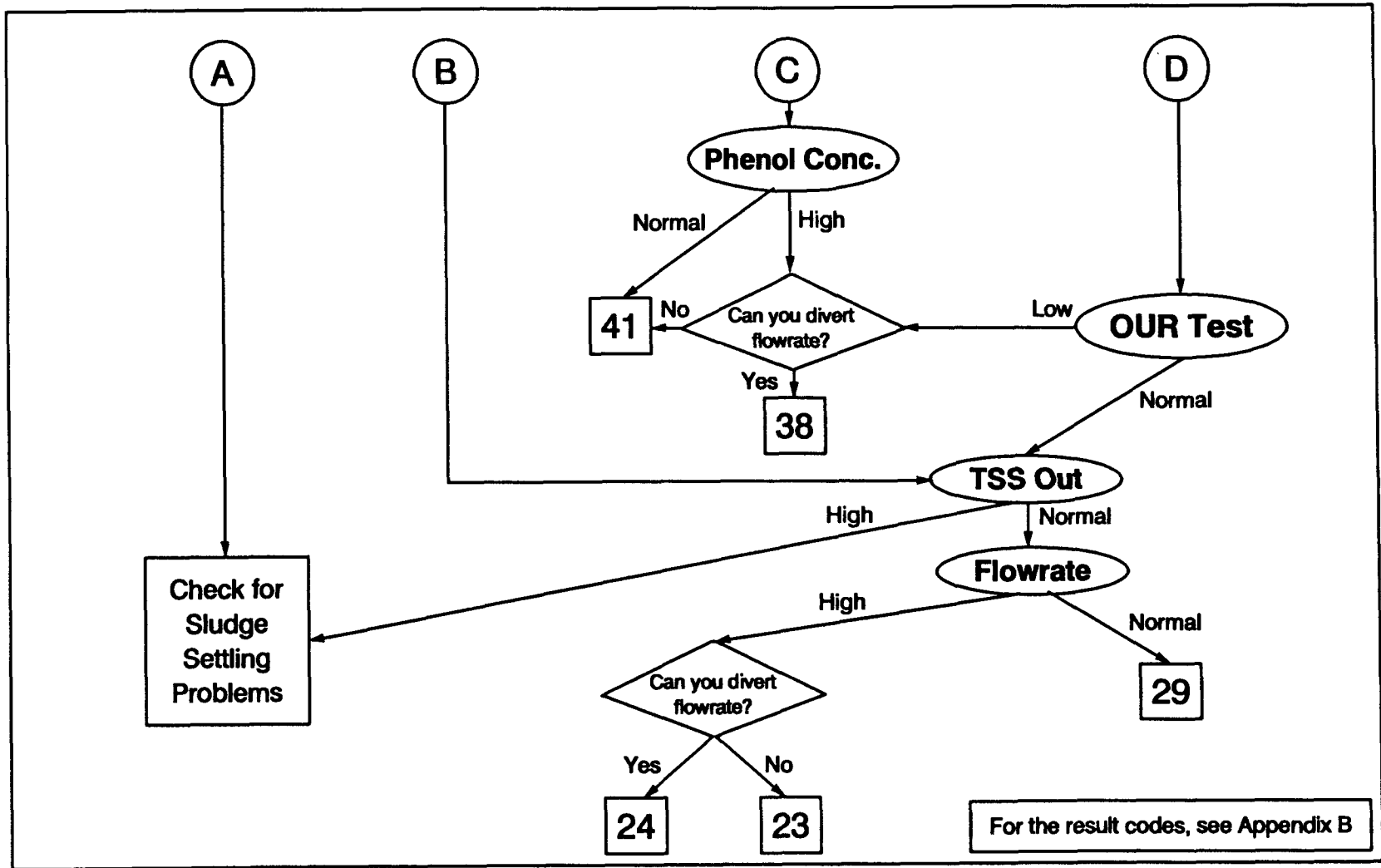


Figure 4-5(b): The sludge age process control logic tree, part 2

MLSS cannot be controlled directly, the sludge wasting rate and retention time in clarifiers affect the sludge MLSS concentration.

When sludge settles in sedimentation tanks, it tends to create a different phase with the water. The solid phase is commonly called the sludge blanket. The sludge blanket depth is another parameter which influences the process control of sludge age. The sludge blanket is monitored by sludge level indicators and can be controlled by both the return sludge and the sludge wasting rate. The level of the sludge blanket should be kept within the upper and lower limits to ensure optimal plant performance.

The sludge wasting rate is the key parameter which controls the sludge age value. Wasting sludge causes reduction of the sludge inventory in the system and results in lower sludge age. Increasing the sludge age involves reduction of the wasting rate as well as the condition of the sludge blanket depth. When the sludge wasting is stopped, the sludge blanket should be monitored closely because solids may overflow into the effluent as a result of high sludge blanket levels.

The sludge wasting is controlled by the WAS pump controls which consists of an ON/OFF switch for the pump and a discharge valve which regulates the amount to be wasted. Sludge flow is monitored by a gauge which reads the volume of sludge wasted per minute. If it is needed, the sludge is wasted based on the wasting flowrate and the time of waste. The waste sludge is pumped to the aerobic sludge digester where it resides until it is discharged into the sanitary sewer.

The return sludge rate is one of the key parameters which influences the MLSS in the aeration basins. The sludge is returned from the sedimentation tanks to supply a sufficient amount of bacteria mass in the activated sludge basins to consume the incoming pollutants. The sludge is returned by pumps which are controlled by ON/OFF switches and discharge valves to adjust the amount of sludge to be returned. A flowmeter is placed on the return sludge pipeline to read the RAS flowrate which records the volume of sludge returned per minute.

The optimal RAS rate is calculated based on three different assumptions given in section 2.1.4.2. and Eq. 2-33 through 2-35. The optimal RAS value is determined by the equation which requires the highest RAS rate among those three equations. However, if the determined RAS rate is lower than the default low RAS limit (40%), the determined value is replaced with the default value and it is assumed equal to the optimal RAS rate. Once the optimal RAS is determined, the current value is examined and ASPX advises the operators to adjust the RAS rate according to the optimal value.

The OUR test is an indicator test of the presence of any toxic chemical in the water. If the test detects the presence of a toxic chemical, ASPX advises taking corrective results to eliminate the toxic chemical source. The test result is compared to the default minimum OUR test result to indicate whether the test result is acceptable.

The food-to-microorganism ratio is also used during the sludge age process control scheme because F/M ratio provides a ratio which can regulate the amount of microorganisms present for particular organic loading. The concentration of organics in the water is measured by a TOC analyzer and is used in F/M calculations (see Eq. 2-4). The MLSS is adjusted to obtain a desired F/M ratio based on the incoming organic loading by adjusting the return activated sludge rate.

4.3.3 Dissolved Oxygen Control by ASPX

Dissolved oxygen is a fundamental requirement for bacteria to survive in the aeration basins. For the nitrification process, a high concentration of dissolved oxygen is essential because the existence of the nitrifying bacteria greatly depends on the dissolved oxygen concentration. Therefore, the oxygen level can be used as an indication of possible failure in the nitrification process.

The DO level in the aeration basins is influenced by a number of parameters including organic loading, and the MLSS concentration. The dissolved oxygen level is monitored by the DO probes which continuously read the dissolved oxygen concentration in the aeration basins. The DO is controlled by the amount of air supplied by the blowers.

The process control logic tree for dissolved oxygen is shown in Figure 4-6(a) and 4-6(b). The control of DO involves F/M ratio, MLSS

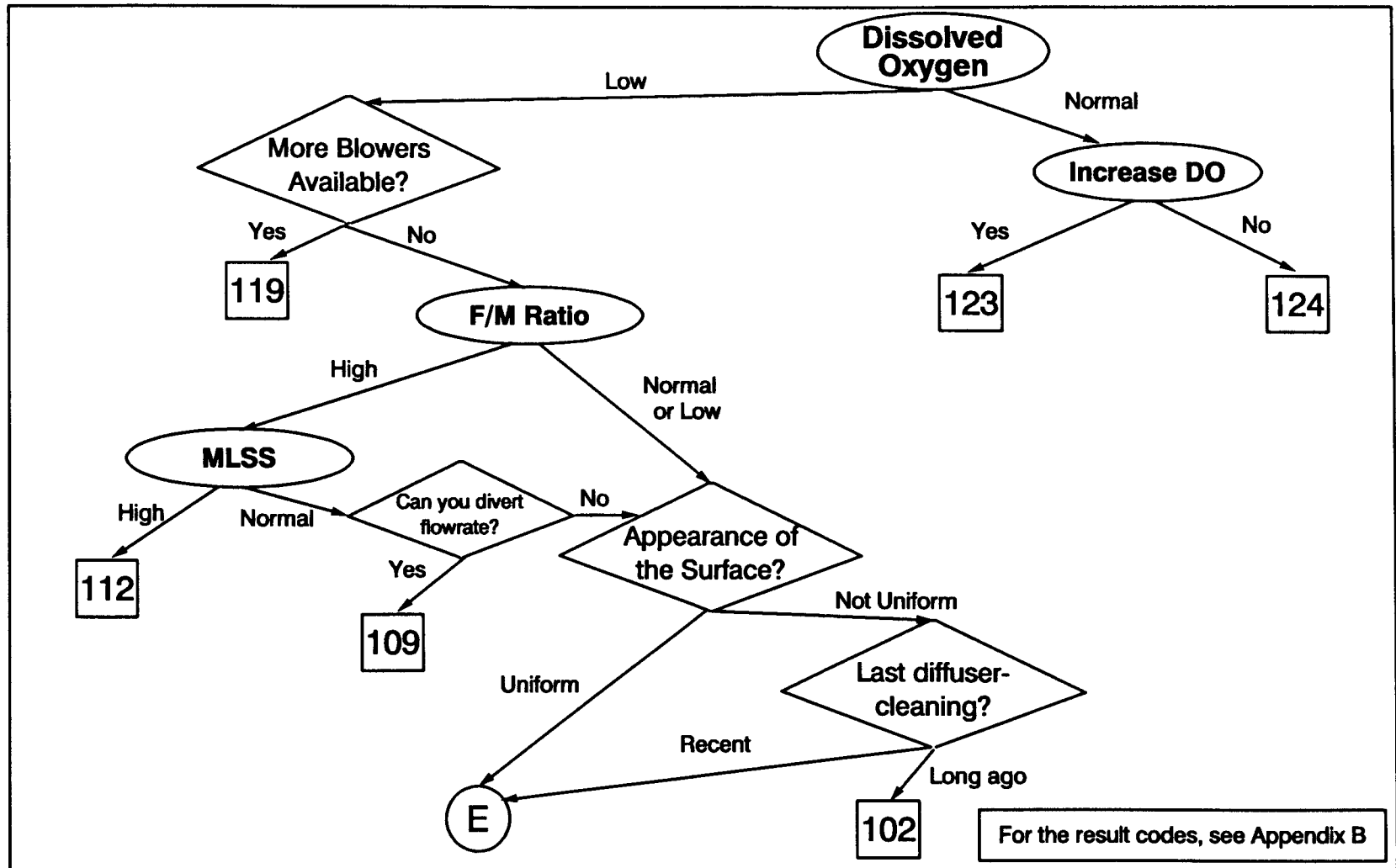


Figure 4-6(a): The dissolved oxygen process control logic tree, part 1

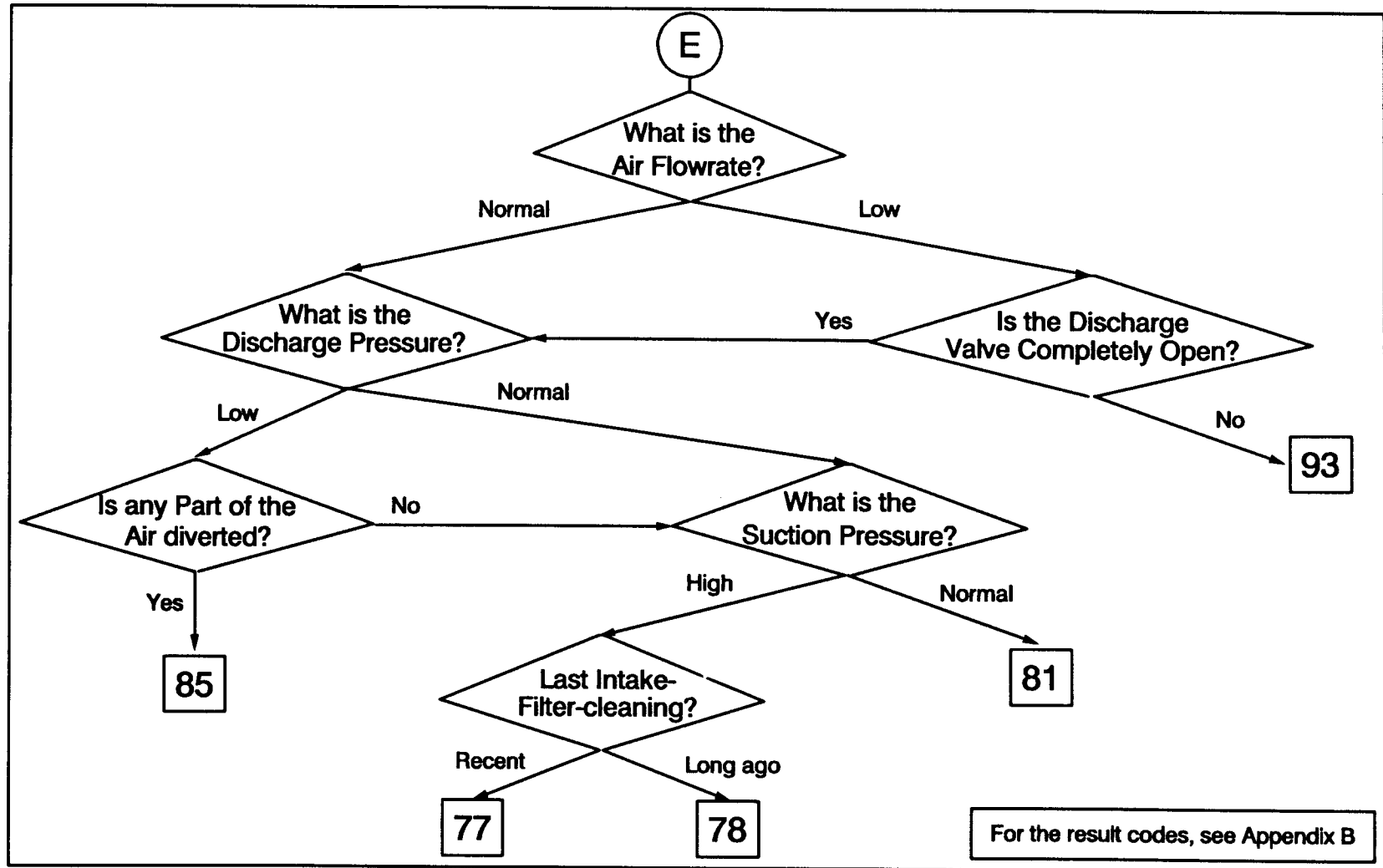


Figure 4-6(b): The dissolved oxygen process control logic tree, part 2

concentration, condition of the aeration tank surface and the diffusers, and aeration related equipment such as air flowrate, discharge valve positioning, discharge and suction pressure gauge reading, and condition of the intake air filters.

The F/M ratio is used as a representative value for the existence of high organic loading within the DO control scheme. If the F/M ratio is high, the DO level would likely be low because the DO is rapidly consumed due to a high organic concentration in the influent. In contrast, when the F/M ratio is low, it indicates that the low DO is caused by something else rather than either high organic loading or improper MLSS concentration.

If the F/M ratio is high, the MLSS concentration is checked to locate the source of the problem. If the MLSS concentration is found to be low, the RAS rate is adjusted accordingly to achieve the desired MLSS concentration to satisfy the optimum F/M value.

The appearance of the aeration basin surface is an indicator of a problem with the blowers or the diffusers. The surface of the basins under normal operating conditions should look uniform with adequate turbulence. If the surface does not look uniform, the conditions of the diffusers need to be considered. The diffusers should be cleaned periodically, perhaps every six months to eliminate clogging of the diffuser pores. One indication of clogged diffusers is when one section in the aeration basin shows blowoff of excessive turbulence. Since the selected refinery plant is equipped with

a fine-bubble diffuser system, more frequent cleaning may be required due to troublesome characteristics of these type of diffusers.

In addition to the organic loading, MLSS, and the condition of the surface, the amount of air supplied to the aeration basins strongly affects the DO level. Each blower has a maximum limit of supplying air which is determined by the capacity of the blower. The amount of air is measured by discharge gauges and expressed in a Standard Cubic Foot per Minute (SCFM) unit. Control of the air flowrate is achieved by throttling the discharge valve which releases the desired amount of air flow into the aeration basins. A rough estimation of the air flowrate is determined by the following equations:

$$K_L a_1 (C_{\infty}^* - DO_1) = K_L a_2 (C_{\infty}^* - DO_2) \quad (4-2)$$

$$K_L a_2 = K_L a_1 \left(\frac{C_{\infty}^* - DO_1}{C_{\infty}^* - DO_2} \right) \quad (4-3)$$

where $K_L a_x$ = Volumetric mass-transfer coefficient for DO_x , 1/h
 C_{∞}^* = Saturation concentration of oxygen, mg/L
 DO_x = Dissolved oxygen concentration at time x, mg/L

Assume: $K_L a_x = Q_{K_L a_x} \cdot \text{Air Flowrate}_x \quad (4-4)$

$$\text{Air flowrate}_2 = \frac{Q_{KLa_1}}{Q_{KLa_2}} \text{ Air flowrate}_1 \left(\frac{C_{\infty}^* - DO_1}{C_{\infty}^* - DO_2} \right) \quad (4-5)$$

where Q_{KLa_x} = Site-specific coefficient for $K_L a_x$
 Air Flowrate_x = Air flowrate needed to reach DO_x, SCFM

The discharge and suction pressure gauges provide an indication of failure in the air supply system. The pressure reading can be correlated to the amount of air volume being supplied. This correlation is affected by the intake air temperature and the barometric pressure. The higher the temperature and the lower the barometric pressure, the less the discharge pressure will rise and the lower discharge pressure means less power will be required to delivery the same amount of air. Therefore, the operations of a blower during winter and summer will be different.

The condition of intake filters can be monitored by the suction pressure gauge reading. High suction pressure indicates a clogged air intake filter which needs to be replaced. Every six to eight months the intake filters are examined and the clogged ones are replaced.

The maintenance of blowers is a significant part of the activated sludge operations. Adopting a proper maintenance schedule eliminates most problems with the blowers such as clogged filters, low performance, loud noise, and excessive vibration. Blowers are one of the vital components of an activated sludge unit which should always be kept at their highest performance levels.

The principal process control equipment for the control of DO are the air blowers. Each blower is controlled by an ON/OFF switch and a discharge valve. Once the single-speed blower is running, the amount of air can be controlled by throttling the discharge valve to the desired position. Roughly, the air flowrate is estimated by Eq. 4-5 to reach a desired dissolved oxygen level. For high air flowrate requirements, more than one blower may be needed to supply the required amount of air to the aeration basins. Low air flowrates may be supplied by one blower with a blow-off valve which is installed on the blower's discharge side to vent a portion of the air to eliminate surging.

4.3.4 pH Control by ASPX

Bacteria, especially nitrifiers can only survive at a certain pH limit which is a narrow margin between 7.5 - 8.6 with an optimum of 7.6. The chance of survival of the nitrifying bacteria out of this pH range is negligible. Furthermore, satisfying the pH limit during the nitrification process is more difficult since the nitrifying bacteria depletes the alkalinity of the water by making pH more unstable. Thus, the control of pH is essential to assure the completion of the nitrification process without problems.

The control of pH is performed based on the logic tree shown in Figure 4-7. The process control parameters involving pH level control include the caustic addition rate, and the influent pH level. The pH levels in

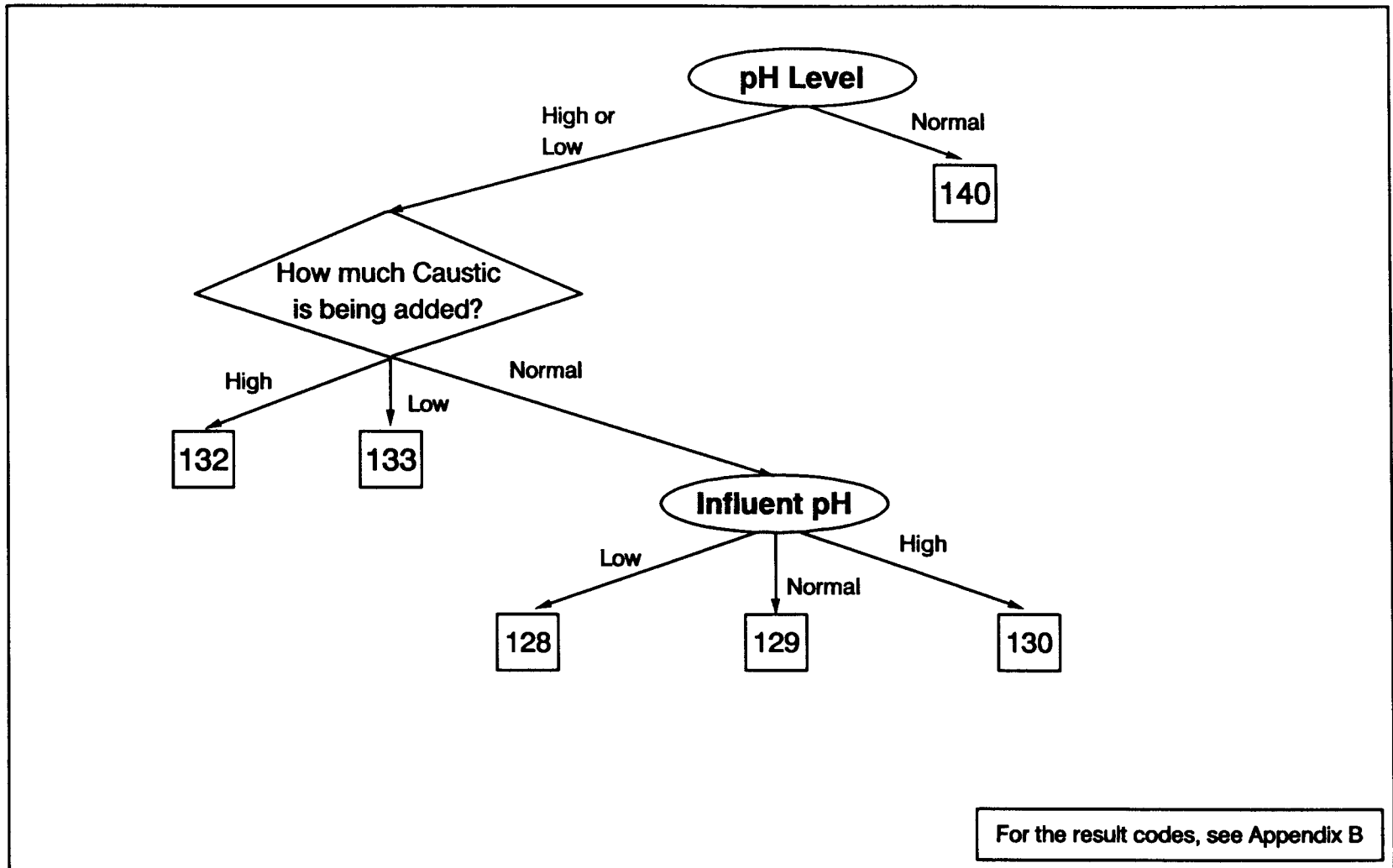


Figure 4-7: The logic tree for the process control of pH level

various locations are monitored by pH electrodes which are placed throughout the plant layout. Both the caustic addition rate and the influent pH level are used during the problem assessment process to locate the source of pH problems.

The caustic addition pipelines are installed at the head end of each aeration basin to release the caustic solution into the basins where the water is being mixed sufficiently. The amount of caustic is controlled by a metering pump which can deliver a specified volume of solution within specified time. The delivery process may be confirmed visually by a mechanical level indicator which is installed on the caustic tank. The required amount of caustic is roughly estimated based on the following equation:

$$Caustic_2 = Caustic_1 \left[\frac{Q_2 C_{NaOH_2} (7.2 - Influent\ pH_2)}{Q_1 C_{NaOH_1} (7.2 - Influent\ pH_1)} \right] \quad (4-6)$$

where $Caustic_x$ = Required caustic rate for Q_x , gpm

Q_x = Flowrate at time x, gpm

$C_{NaOH,x}$ = Concentration of caustic solution at time x, %

Influent pH_x = Influent pH level at time x

Monitoring the influent pH level improves the problem diagnosis process since the indication of abnormal pH levels may be correlated to the nitrification failure. Furthermore, by monitoring the influent pH levels, the treatment plant operators can notify the refinery management about possible acid/base leaks within the refinery processes.

The process control equipment for the control of pH is the caustic addition pumps. The pump controls consist of an ON/OFF switch, and a metering system which is able to deliver a certain volume of caustic to each aeration basin. The required amount of caustic addition is calculated by Eq. 4-6 and delivered by the metering pumps from the chemical storage tanks to the aeration basins through caustic delivery pipelines.

The pump operations are verified by observing the level which is located on the caustic storage tank. The level is recorded daily to calculate the volume of caustic delivered during the past day. The volume of solution is calculated by the following equation:

$$V_{NaOH} = 7.48 \pi r_c^2 (Level_{c,1} - Level_{c,2}) \quad (4-7)$$

where V_{NaOH} = Volume of caustic delivered during time₁-time₂, gal
 r_c = Radius of caustic storage tank, ft
 $Level_{c,x}$ = Level of caustic solution in tank at time x, ft

Confirming the caustic addition operations by the tank level indicator may reveal further problems with the delivery pump and pH electrodes that would have never been detected otherwise.

4.3.5 Phosphate Control by ASPX

Maintaining a sufficient amount of phosphate in the aeration basins for nitrifying bacteria is imperative to achieve a complete nitrification process

without problems. The growth of bacteria is highly dependent on a phosphate source. For proper operations of the activated sludge unit, the ratio, $BOD_5|N|P$ should be equal to $100|5|1$. Since the refinery waste process-water has a deficit of phosphate, an external addition of phosphate is required for the selected refinery treatment plant.

The control of phosphate is performed by the logic tree shown in Figure 4-8. Phosphate control contains one process control parameter which is the rate of phosphate addition. Phosphate is monitored by the effluent phosphate concentration which indicates the remaining portion of the phosphate that was not used by the bacteria in the system. If the effluent phosphate concentration is below limits, the phosphate addition rate is found to be insufficient. For proper operations, the effluent phosphate should be more than the allowed minimum limit to ensure sufficient phosphate for the bacteria to consume. However, an excessive addition of phosphate will only increase the operation costs and will not improve the performance (WPCF, 1987).

Phosphate is added in phosphoric acid form at the head end of the equalization tank. The phosphoric acid is kept in a storage tank which is equipped with a mechanical level indicator similar to the caustic tank. The metering pump for the phosphoric acid delivers the desired volume of acid to the influent of the equalization basin. The pump has an ON/OFF switch and a metering system which adjusts the amount of acid flow delivered to the system. The phosphoric acid flowrate is displayed by a gauge which is installed on the discharge side of the pump. The required amount of

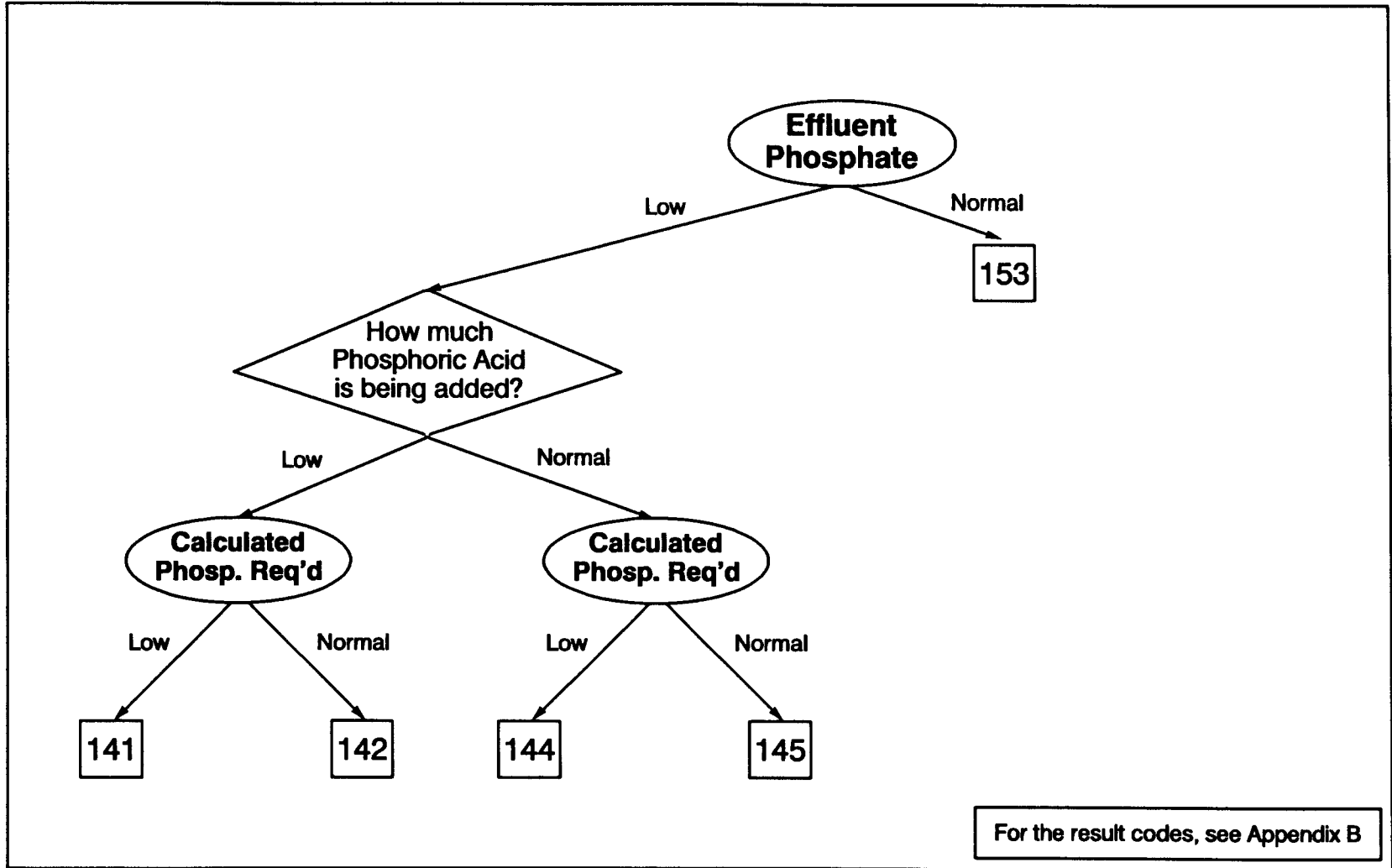


Figure 4-8: The effluent phosphate process control logic tree

phosphoric acid is roughly estimated by the following equations:

Given:

$$\text{Desired BOD}_5/\text{P ratio} = 100/1$$

$$\text{Phosphate content, H}_3\text{PO}_4/\text{P} = 98/31 = 3.2$$

$$\text{Phosphoric acid, density} = 0.96$$

$$\text{purity} = 75\%$$

$$PO_4-P_x, \text{mg/L} = \frac{BOD_{5,x}}{100} \quad (4-8)$$

$$BOD_{5,x} = 1.5 TOC_x \quad (4-9)$$

$$PO_4-P_x, \text{mg/L} = \frac{TOC_x}{66.7} \quad (4-10)$$

$$PO_4-P_x, \text{kg/day} = \frac{(PO_4-P_x, \text{mg/L})(Q_x)(24)(60)(3.7854)}{1000} \quad (4-11)$$

$$PO_4-P_x, \text{gpm} = \frac{(PO_4-P_x, \text{kg/day})(3.2 H_3PO_4/P)}{\left(\frac{75\%}{100\%}\right)(0.96)(24)(60)(3.7854)} \quad (4-12)$$

$$\{H_3PO_4\}_x, \text{gpm} = \frac{\left(\frac{TOC_x}{66.7}\right)(Q_x)(3.2)}{\left(\frac{75\%}{100\%}\right)(0.96)(1000)} \quad (4-13)$$

where PO_4-P_x = Required phosphate at time x
 $BOD_{5,x}$ = BOD_5 concentration at time x, mg/L
 TOC_x = TOC concentration at time x, mg/L

$$Q_x = \text{Flowrate, gpm}$$

$$\{H_3PO_4\}_x = \text{Required amount of phosphoric acid at time x, gpm}$$

Similar to the pH control, the phosphate addition operation is verified by the phosphoric acid level indicator. The following equation is used to calculate the volume of phosphoric acid which is added to the water during a given period:

$$V_{H_3PO_4} = 7.48 \pi r_p^2 (Level_{p,1} - Level_{p,2}) \quad (4-14)$$

where $V_{H_3PO_4}$ = Volume of phosphoric acid delivered during time₁-time₂, gal

r_p = Radius of phosphoric acid storage tank, ft

$Level_{p,x}$ = Level of phosphoric acid in tank at time x, ft

The comparison between the delivered volume of phosphoric acid and the effluent phosphate concentration helps to disclose inconsistencies in the estimation of the phosphate addition rate, and possible errors made measuring phosphate concentration.

4.3.6 Temperature Control by ASPX

Temperature has a significant effect on the nitrification process. The growth kinetic coefficients for the nitrifying bacteria vary greatly with the changing temperature. The colder the temperature, the lower the growth

rate is observed. For a proper nitrification process, the temperature should be kept within the limits at which the nitrifying bacteria grows optimally.

The temperature is controlled by the logic tree shown in Figure 4-9. The primary process control parameter for controlling temperature is the temperature of the influent process-water. The temperature probes are placed at various locations in the plant, including the influent, to monitor the temperature value around the plant. The influent temperature is used to locate the source of the low temperature problems. If the influent temperature is low, it is an indication that there might be a merger of cold-water into the process-water within the refinery processes. Elimination of the merging of cold-water would solve the problems which are caused by low temperatures.

If the water temperature is low due to the weather conditions, there is no direct control of temperature in practice. Few limited regulating actions may be taken under such conditions including directing part of the influent flow into holding tanks, increasing the dissolved oxygen concentration, or increasing the sludge age.

At low temperatures, the growth of bacteria is very limited, thus, less organic load can be consumed by the bacteria mass. The incoming organic load can temporarily be reduced by diverting part of the influent flow into the holding tanks. This will only reduce the load temporarily until a higher temperature is obtained. That part of the water which is being held

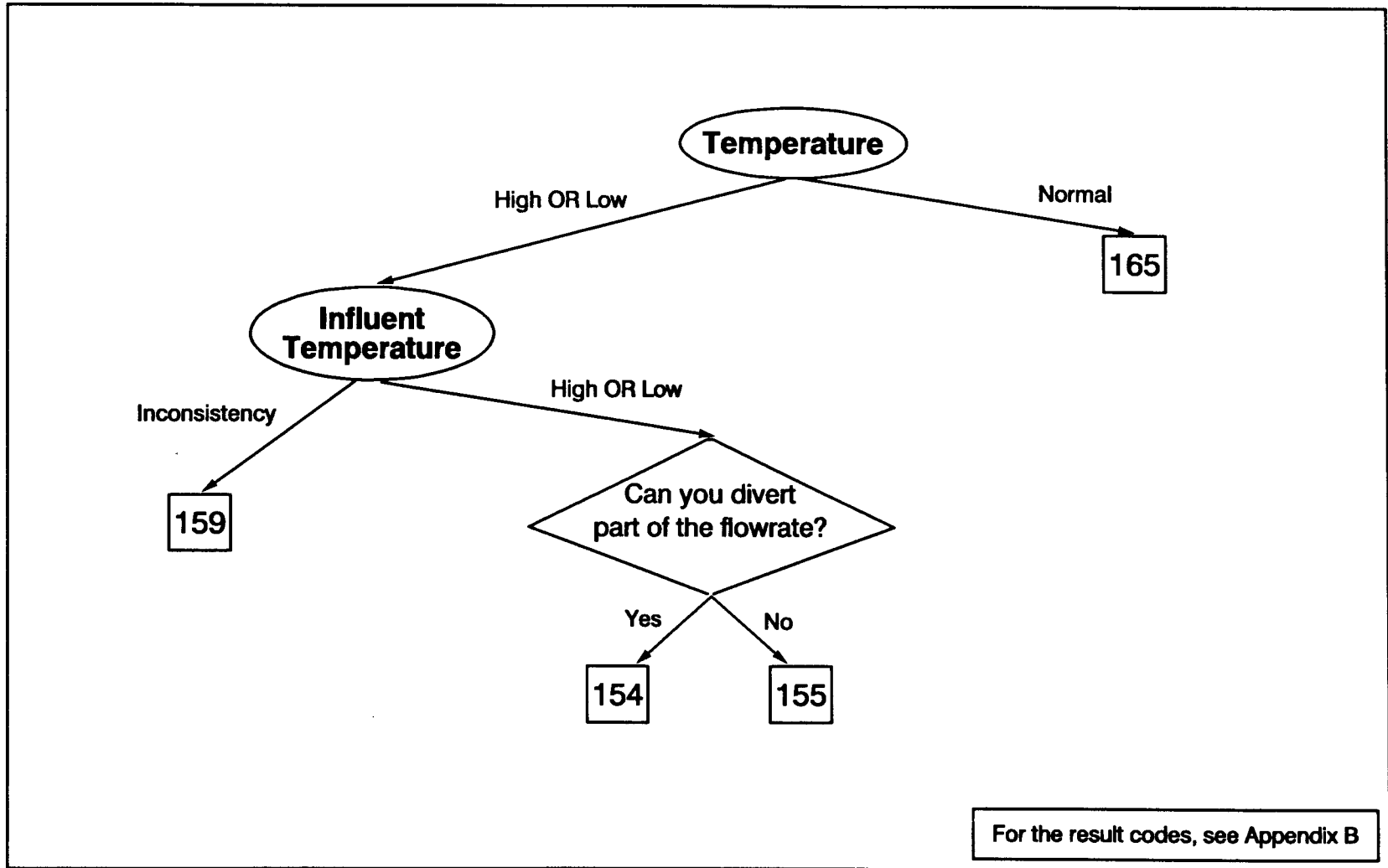


Figure 4-9: The logic tree for the process control of temperature

in the holding tanks should be released back to the influent of the plant as soon as possible to avoid problems caused by an excessive holding time. Increasing the dissolved oxygen concentration would decrease the risk of losing the nitrification capability due to low temperatures. The DO should be increased even though the oxygen level is within the acceptable limits. At low temperatures, supplying oxygen may be difficult because the drive which forces the oxygen transfer is lower due to the lower saturation concentration for oxygen at the lower temperatures. Furthermore, low temperatures will cause lower air density and higher power demand from the air blowers.

To ensure the occurrence of the nitrification process, sludge age may be increased when the low temperatures are experienced in the plant. Sludge wasting is reduced or stopped depending on the sludge blanket depth in the sedimentation tanks. Increased sludge age may eliminate problems with the nitrification process caused by low temperatures if the temperature drop is not severe.

4.4 Future uses of ASPX in the selected plant

Currently, ASPX is designed as a diagnostic tool for the nitrification problems in the selected refinery plant. An operator can use ASPX to identify nitrification upsets within the plant. During the computer consultation, the historical data base is analyzed to obtain a data pattern and statistical information. At the end of the consultation, the computer

displays the detected upsets, if any, and includes suggestions to correct the problems. "Why" and "how" explanations are also included to inform the user about the computer's reasoning methodology.

Possibilities for the future applications of ASPX in the selected plant are numerous. Some such possibilities include the comprehensive coverage of activated sludge process problems including sludge bulking, use of ASPX as a trainer for the operators, and applying ASPX as a plant simulator.

Comprehensive coverage. Future work on ASPX will expand beyond the nitrification process reaching some of the most common problems experienced in the activated sludge process. Sludge bulking is one of these problem areas which will receive the most attention since the occurrence of bulking problems is overwhelming (see section 2.1.3.4). The bulking problems are not included in the original ASPX program because there are commercially available expert systems that cover the bulking process extensively (Gall and Patry, 1988).

ASPX may be expanded to other sludge settlement problems such as floating sludge, pinpoint floc, ashing, and straggler floc (see section 2.1.3.5). Elimination of sludge settlement problems improves the effluent water quality and the overall plant performance.

Training tool. One of the most important tasks for the future application of ASPX is to act as a trainer for the plant operators. ASPX can be an invaluable opportunity for the trainee operators to acquire practical

experience. Using expert systems as a training tool is a fast, safe, and an effective way to obtain the practical experience which would have otherwise taken considerable time to acquire.

Generally, the operators test their knowledge by experimenting with the real-time processes. The trial and error method is used often to test validity of a decision. However, the trial and error method greatly endangers the environment because of high risks involved with the method due to uncertainties in knowing the outcome.

The use of the ASPX expert system training tool, in contrast, will be a safe way to practice. The ASPX expert system can create a teaching environment which is capable of teaching the "practical experience" without the unsafe real-time trial and error attempts with the plant's processes.

The ASPX training tool will create hypothetical situations and present them to the operator as if they were real-time applications. ASPX will simulate an upset and will expect the operator to identify the source of the upset and to suggest several corrective actions. If the operator's judgement is correct then the operator will be rewarded. But if the judgement leads to a disaster, the operator will be advised to review the suggested corrective actions. If the operator cannot identify the correct action successfully, the trainer will give the answer to the operator. Use of this method in operator training courses will help trainee operators to learn practical

aspects of the activated sludge process before they start their job as certified plant operators.

Training sessions may be made more attractive by the use of multi-media technology. Today's computers are capable of teaching operators interactively through the motion picture adaptation of the activated sludge process control techniques. The use of audiovisual multi-media applications will make the teaching sessions more attractive and more effective. A walk-through exercise of the plant will provide a visual contact with the real-time processes. This exercise will include an audio explanation session which eliminates extensive text material so the operators can focus on the content. The training sessions with the multi-media systems will be more realistic than the text book cases and more fun than sitting in a classroom.

Process simulator. Although many mathematical models for the wastewater treatment plant processes are available, the activated sludge process has not been completely simulated, yet. Future development of the ASPX simulator will provide an inside look at the activated sludge process by the information supplied by both the mathematical models and the knowledge of expert persons in the wastewater treatment plant operations field. Operators will be able to use the simulator to understand the behavior of the activated sludge process. They will be able to test the process control equipment adjustments and the impact of each adjustment on the overall plant performance, as well as finding an optimal rate for the addition of each chemical.

The ASPX process simulator of the future will enable the program to be used as a forecasting tool. The operators will enter hypothetical conditions and the expert system will predict the future conditions based on the given facts. This method will be useful for the operator to learn more about the behavior of the activated sludge process. With help from the simulator, the operator will acquire the necessary skills to predict the performance of the activated sludge process under different conditions. The more the operator tries various conditions, the more he/she will learn about the behavior of the process.

The process simulation mode of ASPX will provide operators with the ability to test the process control equipment adjustments. Although there is a massive amount of literature present pertaining to the activated sludge process control equipment, the information is a limited source for understanding the significance of each control equipment adjustment on real-time applications. Common questions like "Should I put the second blower in service so that the DO will rise to 3 mg/L?" or "What is the optimum sludge return rate?" will be able to be tested with the process simulation module. By simulating under different settings of the equipment, the optimum adjustment required will be able to be identified. The testing equipment adjustment with the simulator will eliminate the real-time trial and error method by reducing the risk of violating the NPDES permit.

Process simulation will allow operators to understand the changes in the activated sludge process due to the adjustments which will be made during

the simulation mode. Practicing with the simulator will give the operator a feeling for the magnitude of the impact caused by the process control equipment. Understanding the impact of individual equipment on the overall plant performance will provide operators with experience in controlling the plant's equipment.

Testing other settings such as determining an optimum rate for the caustic addition to obtain a certain pH will be able to be simulated by the process simulation module. The determination process will be performed under the simulation mode by the trial and error method and by observing the outcome of the simulation. Finding the optimum rate for the chemical addition provides for reduced operational costs and less chance of problems with the activated sludge process.

Process simulation will also be used to verify the answers given by the diagnostic module of ASPX. Before the ASPX program makes a suggestion to the operator, the program will verify the action by internally simulating the suggestion. The program will assume the suggested action is correct and apply this action to the simulator to obtain an outcome. The obtained outcome from the simulation will then be compared with the desired outcome. If the desired outcome is reached based on the simulation, the suggested action will be strongly recommended. In contrast, failure to reach the desired outcome will indicate less confidence with the suggested action.

After an adequate amount of practice with the simulator, the operator will gain practical experience, an extensive understanding of the activated sludge process, an ability to comprehend consequences of adjustments made on the process control equipment, and an ability to use the process control equipment more effectively.

5 Conclusions

This dissertation applied newly developed expert system technology to diagnose and correct nitrification problems within the selected refinery wastewater treatment plant. The primary goal of the *Activated Sludge Process eXpert* (ASPX) is to diagnose and correct problems with the nitrification process at a level comparable to that of a senior operator who has 20-years of practical experience in activated sludge operations.

The development of the knowledge-based system consisted of the following procedures. First, knowledge was compiled to diagnose the problems and to control the nitrification process in the selected refinery plant. Second, the logic trees and the rules were created for the six selected process control parameters which affect the nitrification process most significantly. Third, the data base functions including the statistical and mathematical models were developed to obtain information about the historical and the future conditions of the plant. Fourth, the user-interface which enables easy access to full functions of the ASPX program without the need for memorizing command statements was created. Finally, implementation and testing of the knowledge-based system was performed using both hypothetical problem cases and full scale plant data.

The knowledge base consists of six frames including the frames controlling the mean cell retention time, pH, dissolved oxygen, temperature, ammonia levels, and phosphate concentration. Each parameter has a logic tree which was developed to diagnose any problems associated with that

parameter. Corrective suggestions were also included in the logic tree structure for the solution of the detected problem. The knowledge base consists of a total of 337 rules and 173 suggestions which were created based on the logic trees.

The data base functions include data entry forms which allow the operator to enter the results of the daily process monitoring. Graphical representation and statistical information of the data base records are also made available to the operators. Statistical and mathematical models were developed to analyze the data base records. The statistical model is used to perform two functions. First, it is used to define fuzzy terms such as "low" and "high." Second, the Bayesian statistical model is used to predict the minimum required mean cell retention time to achieve a desired nitrification rate without violating the effluent ammonia standard. The mathematical model was developed to predict effluent ammonia concentration as a function of the plant's current and future conditions. The mathematical model also estimates the minimum mean cell retention time which provides an effluent ammonia concentration meeting the NPDES permit discharge limits.

The user-interface was created from scratch to enable operators unfamiliar with computers to effectively use the program. The user-interface was written in Pascal and it consists of "Mac-like" menu selections and powerful mouse control commands. The operator is only required to know basic access commands in order to use the program's full capabilities. The user-interface module is an essential part of developing a knowledge-based

system because many operators refrain from using the computer due to a lack of experience or fear.

The testing procedure was divided into three steps which include the initial, intermediate, and advanced testing procedure. The initial testing was performed by the developer who analyzed the rules and suggestions given by the program. Hypothetical cases were presented to the program and suggestions were analyzed by the developer. Any inconsistency within the results were corrected by either modifying the rules or by adding additional rules into the knowledge base.

After the initial testing, the program and computer must be installed at the treatment facility. The computer must be physically set up in the control room at the selected refinery plant where the intermediate testing is performed using real-time data base values during the real-time process, and the diagnosis of nitrification problems are made. Advanced testing is the final step in the testing procedure. The program must be applied to real-time operations where it will make process control decisions using real-time plant data and the rules in its knowledge base. If the program gives an erroneous suggestion, the rules will be modified until all inaccurate suggestions are eliminated.

Future work needed for the application of ASPX may include the following improvements:

- More comprehensive coverage of the activated sludge process such as bulking, foaming may be included in the diagnostics section.

- The development of a trainer module used for educating the plant operators may also be included within the ASPX program.
- The development of a more detailed mathematical model may be included to simulate the plant processes and to be used as an optional process control technique.

The knowledge-based expert system developed in this dissertation is the first knowledge-based system specifically developed for the process control of nitrification in refinery treatment plants. Combining the statistical, mathematical, and expert persons' knowledge allows the developer to create a multi-functional knowledge-based system which can be used as a diagnostic tool as well as a process control technique. The development of ASPX is a further step in reaching an optimum process control strategy for the activated sludge process operations. This step will help reveal more about the future of convergence in the fields of expert systems and wastewater engineering. Future developments of ASPX will open a new door into the exciting world of activated sludge process.

References

- Abelson, H., Sussman, G. J., and Sussman, J. (1985). *The structure and interpretation of computer programs*. Cambridge, MA: MIT Press.
- Abufayed, A. A., and Schroeder, E. D. (1986). Kinetics and stoichiometry of SBR/denitrification with a primary sludge carbon source. *J. Water Pollut. Control Fed.*, 58(5), 398-405.
- Aleem, M. I. H., Hoch, G. E., and Varneo, J. E. (1965). Water as the source of oxidant and reductant in bacterial chemosynthesis. *Proc. Nat. Acad. Sci. U.S.*, 54, 869-873.
- Alty, J. L., and Coombs, M. J. (1984). *Expert systems: Concepts and examples*. Manchester, England: NCC Publications.
- Andrews, J. F. (1974). Dynamic models and control strategies for wastewater treatment processes. *Water Research*, 8, 261-289.
- Anthonisen, A. C., Loehr, R. C., Prakasam, T. B. S., and Srinath, E. G. (1976). Inhibition of nitrification by ammonia and nitrous acid. *J. Water Pollut. Control Fed.*, 48(5), 835-852.
- Antoniou, P., Hamilton, J., Koopman, B., Jain, R., Holloway, B., Lyberatos, G., and Svoronos, S. A. (1990). Effect of temperature and pH on the effective maximum specific growth rate of nitrifying bacteria. *Water Research*, 24(1), 97-101.
- APHA, AWWA, and WPCF (1989). *Standard methods: For the examination of water and wastewater, (17th Ed.)*. Washington, D. C.: American Public Health Association.
- Argaman, Y., and Brenner, A. (1986). Single-sludge nitrogen removal: Modeling and experimental results. *J. Water Pollut. Control Fed.*, 58(8), 853-860.

- Armijos, A., Wright, J. R., and Houck, M. H. (1990). Bayesian inferencing applied to real-time reservoir operations. *J. of Water Resources Planning and Management*, 116(1), 38-51.
- Atkins, J. S., Kunz, J. C., Shortliffe, E. H., and Fallat, R. J. (1984). PUFF: An expert system for interpretation of preliminary function data. In B. C. Clancey and E. H. Shortliffe (Eds.), *Readings in medical artificial intelligence: The first decade*. Reading, MA: Addison-Wesley.
- Awano, T., Kamitani, H., Kimura, T., Noda, S., and Maruyama, H. (1990). A pump control system at a sewage treatment plant, using an AI approach. In R. Briggs (Ed.), *Instrumentation, control and automation of water and wastewater treatment and transport systems. Proceedings of the 5th IAWPRC Workshop held in Yokohama and Kyoto, Japan 26 July-3 Aug., 1990*. Oxford, England: Pergamon Press. 721-726.
- Baba, K., Enbutu, I., Matuzaki, H., and Nogita, S. (1990). Intelligent support system for water and sewage treatment plants which includes a past history learning function--coagulant injection guidance system using neuralnet algorithm. In R. Briggs (Ed.), *Instrumentation, control and automation of water and wastewater treatment and transport systems. Proceedings of the 5th IAWPRC Workshop held in Yokohama and Kyoto, Japan 26 July-3 Aug., 1990*. Oxford, England: Pergamon Press. 227-234.
- Balakrishnan, S., and Eckenfelder, W. W. (1969). Nitrogen relationships in biological treatment processes, I. Nitrification in the activated sludge process. *Water Research*, 3, 73-81.
- Barnett, M. W., and Andrews, J. F. (1987). The use of expert systems in computer assisted operation of the anaerobic digestion process. *Paper presented at the Texas Water Pollution Control Association Conference, June 3-5, Corpus Christi, TX*.

- Barnett, M. W., and Andrews, J. F. (1990). Knowledge-based systems for operation of wastewater treatment processes. In R. Briggs (Ed.), *Instrumentation, control and automation of water and wastewater treatment and transport systems. Proceedings of the 5th IAWPRC Workshop held in Yokohama and Kyoto, Japan 26 July-3 Aug., 1990*. Oxford, England: Pergamon Press. 211-218.
- Barnwell, Jr., T. O., and Boynton, D. K. (1985). EPA supported waste load allocation models. In *Proceedings of the Specialty Conference, Computer Applications in Water Resources, Buffalo, New York*. New York, NY: American Society of Civil Engineers. 934-943.
- Barnwell, Jr., T. O., Brown, L. C., and Marek, W. (1986). *Development of a prototype expert advisor for the enhanced stream water quality model QUAL2E*. Athens, GA: U.S. EPA, Environmental Research Laboratory.
- Barr, A., and Feigenbaum, E. A. (Eds.) (1981). *The handbook of artificial intelligence, Vol. 1*. Los Altos, CA: William Kaufman.
- Beaverstock, M., Bristol, E. H., and Fortin, D. (1985). Expert systems as a stimulus to improved process control. In *Proceedings of the 1985 American Control Conference (Cat. No. 85CH2119-6), Vol. 2*. New York, NY: IEEE. 898-903.
- Beccari, M., Marani, D., and Ramadori, R. (1979). A critical analysis of nitrification alternatives. *Water Research*, 13, 185-192.
- Beccari, M., Passino, R., Ramadori, R., and Tandoi, V. (1983). Kinetics of dissimilatory nitrate and nitrite reduction in suspended growth culture. *J. Water Pollut. Control Fed.*, 55(1), 58-64.
- Beck, M. B., Latten, A., and Tong, R. M. (1978). *Modelling and operational control of the activated sludge process in wastewater treatment*. Laxenburg, Austria: International Institute for Applied Systems Analysis. (Professional Paper PP-78-10).

- Beck, M. B. (1981). Operational estimation and prediction of nitrification dynamics in the activated sludge process. *Water Research*, 15(12), 1313-1330.
- Beck, M. B. (1984). Modelling and control studies of the activated sludge process at Norwich Sewage Works. *Trans. Inst. M. C.*, 6(3), 117-131.
- Bennett, J. S., and Engelmores, R. S. (1979). SACON: A knowledge-based consultant for structural analysis. In *Proceedings of the Sixth IJCAI 79 Joint Conference on Artificial Intelligence*. Tokyo, Japan. 47-49.
- Berthouex, P. M., Hunter, W. G., and Pallesen, L. (1978). Dynamic behavior of an activated sludge plant. *Water Research*, 12, 957-972.
- Berthouex, P. M., Lai, W., and Darjatmoko, A. (1989). Statistics-based approach to wastewater treatment plant operations. *J. of Environmental Engineering, American Society of Civil Engineers*, 115(3), 650-671.
- Blockley, D. I. (1987). Uncertainty analysis in expert systems. *Civil Engineering Systems*, 4, 3-6.
- Boger, Z., and Guterman, H. (1990). Knowledge representation of wastewater treatment plant operating history and current state. In R. Briggs (Ed.), *Instrumentation, control and automation of water and wastewater treatment and transport systems. Proceedings of the 5th IAWPRC Workshop held in Yokohama and Kyoto, Japan 26 July-3 Aug., 1990*. Oxford, England: Pergamon Press. 235-242.
- Boon, B., and Laudelout, H. (1962). Kinetics of nitrite oxidation by *Nitrobacter winogradskyi*. *Biochem. J.*, 85, 440-447.

- Bowerman, R. G., and Glover, D. E. (1988). *Putting expert systems into practice*. New York, NY: van Nostrand Reinhold Co.
- Box, G. E. P., and Jenkins, G. M. (1976). *Time series analysis forecasting and control*. San Francisco, CA: Holden-Day.
- Brachman, R. J., and Levesque, H. J. (1985). *Readings in knowledge representation*. Los Altos, CA: Morgan Kaufmann.
- Brachman, R. J., and Schmolze, J. G. (1983). An overview of the KL-ONE knowledge representation system. *Cognitive Sci.*, 9(2), 170-216.
- Brown, C. B. (1987). Jaynes, Bayes, and fuzzy identification. *Civil Engineering Systems*, 4, 142-146.
- Busby, J. B., and Andrews, J. F. (1975). Dynamic modelling and control strategies for the activated sludge process. *J. Water Pollut. Control Fed.*, 47(5), 1055-1080.
- Buswell, A. M., Shiota, T., Lawrence, N., and Meter, I. V. (1954). Laboratory studies in the kinetics of the growth of *Nitrosomonas* with relation to the nitrification phase of the BOD test. *Appl. Microbiol.*, 2, 21-25.
- Chambers, B. (1982). Effect of longitudinal mixing and anoxic zones on settleability of activated sludge. In B. Chambers, and E. J. Tomlinson (Eds.), *Bulking of activated sludge: Preventive and remedial methods*. Chichester, England: Ellis Horwood Ltd. 166-183.
- Chambers, B., and Tomlinson, E. J. (1982). The cost of chemical treatment to control the bulking of activated sludge. In B. Chambers, and E. J. Tomlinson (Eds.), *Bulking of activated sludge: Preventive and remedial methods*. Chichester, England: Ellis Horwood Ltd. 264-270.

- Chan, W. T., and Koe, L. C. C. (1991). A knowledge-based framework for the diagnosis of sludge bulking in the activated sludge process. In P. Grau and the IAWPRC Programme Committee (Eds.), *Proceedings of the fifteenth biennial conference of the International Association on Water Pollution Research and Control, held in Kyoto, Japan 29 July-3 Aug., 1990*. *Water Sci. Tech.*, 23, 847-855.
- Chapman, T. D., Matsch, L. C., and Zander, E. H. (1976). Effects of high dissolved oxygen concentration in activated sludge system. *J. Water Pollut. Control Fed.*, 48(11), 2486-2510.
- Characklis, W. G., and Gujer, W. (1979). Temperature dependency of microbial reactions. In S. H. Jenkins (Ed.), *Prog. Water Tech. Suppl. 1. Kinetics of wastewater treatment*. Oxford, England: Pergamon Press. 111-130.
- Charley, R. C., Hooper, D. G., and McLee, A. G. (1980). Nitrification kinetics in activated sludge at various temperature and dissolved oxygen concentrations. *Water Research*, 14, 1387-1396.
- Chen, C., and Christensen, E. R. (1985). A unified theory for microbial growth under multiple nutrient limitation. *Water Research*, 19(6), 791-798.
- Chu, W. S., and Willis, R. (1984). An explicit finite difference model for unconfined aquifers. *Ground Water*, 22(6), 728-734.
- Chudoba, J., Ottova, V., and Madera, V. (1973a). Control of activated-sludge filamentous bulking--I. Effect of the hydraulic regime or degree of mixing in an aeration tank. *Water Research*, 7, 1163-1182.
- Chudoba, J., Grau, P., and Ottova, V. (1973b). Control of activated-sludge filamentous bulking--II. Selection of microorganisms by means of a selector. *Water Research*, 7, 1389-1406.

- Chudoba, J., Blaha, J., and Madera, V. (1974). Control of activated-sludge filamentous bulking--III. Effect of sludge loading. *Water Research*, 8, 231-237.
- Chudoba, J., Cech, J. S., Farkac, J., and Grau, P. (1985). Control of activated sludge filamentous bulking. *Water Research*, 19(2), 191-196.
- Clayton, J. A., Ekama, G. A., Wentzel, M. C., and Marais, G. v. R. (1991). Denitrification kinetics in biological nitrogen and phosphorus removal activated sludge systems treating municipal waste waters. In P. Grau and the IAWPRC Programme Committee (Eds.), *Proceedings of the fifteenth biennial conference of the International Association on Water Pollution Research and Control, held in Kyoto, Japan 29 July-3 Aug., 1990*. *Water Sci. Tech.*, 23, 1025-1035.
- Clifft, R. C., and Andrews, J. F. (1981). Predicting the dynamics of oxygen utilization in the activated sludge process. *J. Water Pollut. Control Fed.*, 53(7), 1219-1232.
- Clocksins, W. F., and Mellish, C. S. (1981). *Programming in Prolog*. New York, NY: Springer-Verlag.
- Connolly, J. P., and Winfield, R. P. (1984). *A user's guide for WASTOX, a framework for modeling the fate of toxic chemicals in aquatic environments*. Gulf Breeze, FL: U.S. EPA (EPA-600/3-84-077).
- Cumberpatch, J., and Heaps, H. S. (1976). A disease-conscious method for sequential diagnosis by use of disease probabilities without assumption of symptom independence. *International J. of Biomedical Computing*, 7, 61-78.

- Daigger, G. T., Robbins, M. H., and Marshall, B. R. (1985). The design of a selector to control low-F/M filamentous bulking. *J. Water Pollut. Control Fed.*, 57(3), 220-226.
- Danckwerts, P. V. (1970). *Gas liquid reactions*. New York, NY: McGraw-Hill Book Co.
- Dearden, H. (1989). *Application of self-tuning control for process plant optimization*. In Proceedings of the ISA/89 Advanced Control Conference. 4.1-4.14.
- Doneker, R. L., and Jirka, G. H. (1988). *CORMIX1 An expert system for mixing zone analysis of conventional and toxic single port aquatic discharges*. Athens, GA: U.S. EPA, Environmental Research Laboratory (EPA-600/3-88-013).
- Downing, A. L., Painter, H. A., and Knowles, G. (1964). Nitrification in the activated sludge process. *J. Inst. Sew. Purif.*, 2, 130-158.
- Duda, R. O., Hart, P. E., and Nilsson, N. J. (1976). Subjective Bayesian methods for rule-based inference systems. In *Proceedings of the National Computer Conference*. 1075-1081.
- Duda, R. O., Hart, P. E., Barrett, P., Gaschnig, J., Konolige, K., Reboh, R., and Slocum, J. (1978). Development of the PROSPECTOR consultation system for mineral exploration. *Final report, SRI Projects 5821 and 6415, Artificial Intelligence Center*. Menlo Park, CA: SRI International.
- Dyer, J. C., and Mignone, N. A. (1983). *Handbook of industrial residues*. Park Ridge, NJ: Noyes Publications.
- Edmunds, R. A. (1988). *The Prentice Hall guide to expert systems*. Englewood Cliffs, NJ: Prentice-Hall.

- Eikelboom, D. H. (1975). Filamentous organisms observed in activated sludge. *Water Research*, 9, 365-388.
- Eikelboom, D. H. (1982). Biosorption and prevention of bulking sludge by means of a high floc loading. In B. Chambers, and E. J. Tomlinson (Eds.), *Bulking of activated sludge: Preventive and remedial methods*. Chichester, England: Ellis Horwood Ltd. 90-104.
- Eiler, R. G., and Smith, R. (1978). The role of computer programs in preliminary design. In J. W. Male, and S. P. Graef (Eds.), *Applications of computer programs in the preliminary design of wastewater treatment facilities*. Cincinnati, OH: U.S. EPA, Municipal Environmental Research Laboratory.
- Ellis, G. W., Ge, X., and Grasso, D. (1990). Time series analysis of wastewater quality. In R. Briggs (Ed.), *Instrumentation, control and automation of water and wastewater treatment and transport systems. Proceedings of the 5th IAWPRC Workshop held in Yokohama and Kyoto, Japan 26 July-3 Aug., 1990*. Oxford, England: Pergamon Press. 441-448.
- Engel, M. S., and Alexander, M. (1958). Growth and autotrophic metabolism of *Nitrosomonas europaea*. *J. Bact.*, 76, 217-222.
- EPA (1975). *Process design manual for nitrogen control*. Cincinnati, OH: U.S. EPA, Office of Technology Transfer (EPA/625/1-75-007, PB-259 149).
- EPA (1980). *Treatability manual, technologies for control/removal of pollutants, Vol. III*. Washington, D. C.: U.S. EPA (EPA/600/8-80-042c).
- Ernst, G., and Newell, A. (1969). *GPS: A case study in generality and problem solving*. New York, NY: Academic Press.

- Fain, J., Gorlin, D., Hayes-Roth, R., Rosenschein, J., Sowizral, H., and Waterman, D. (1981). *The ROSIE reference manual. Technical Report NI-1647-ARPA*. Santa Monica, CA: Rand Corp.
- Farquhar, G. J., and Boyle, W. C. (1971). Occurrence of filamentous microorganisms in activated sludge. *J. Water Pollut. Control Fed.*, 43(5), 779-798.
- Finger, R. E. (1973). Solids control in activated sludge plants with alum. *J. Water Pollut. Control Fed.*, 45(8), 1654-1662.
- Finn, G. A., and Hall, J. R. (1986). An expert system for rotating equipment vibration analysis. In *Proceedings of the Fourth Conference on Computing in Civil Engineering*. Boston, MA: American Society of Civil Engineers.
- Flanagan, M., and Finnan, K. (1987). Personal computers expand data management capabilities of a distributed digital control system. In *Proceedings of the ISA/87 International Conference and Exhibit: Advances in Instrumentation*, 42(1). Research Triangle Park, NC: ISA. 351-355.
- Fox, J., Barber, D., and Bardhan, K. D. (1980). Alternatives to Bayes? A quantitative comparison with rule-based diagnostic inference. *Meth. Inform. Med.*, 19(4), 210-215.
- Fritsch, T. J. (1986). Pump-Pro: A centrifugal pump diagnosis expert system. In *Proceedings of the 1986 ASME/TEXAS A&M Symposium, Houston, TX*.
- Gall, R. A., and Patry, G. G. (1988). *Knowledge-based systems in wastewater engineering*. McMaster File No. 5-52618, DSS Project: KE405-7-6012/01-SE.
- Gallant, M., Guida, G., Spampinato, L., and Stefanini, A. (1985). Representing procedural knowledge in expert systems: An

application to process control. In *Proceedings of Ninth International Joint Conference on Artificial Intelligence*. 345-352.

Gee, C. S., Pfeffer, J. T., and Suidan, M. T. (1990a). *Nitrosomonas* and *Nitrobacter* interactions in biological nitrification. *J. of Environmental Engineering, American Society of Civil Engineers (ASCE)*, 116(1), 4-17.

Gee, C. S., Suidan, M. T., and Pfeffer, J. T. (1990b). Modeling of nitrification under substrate-inhibiting conditions. *J. of Environmental Engineering, American Society of Civil Engineers (ASCE)*, 116(1), 18-31.

Gee, D. M., and MacArthur, R. C. (1978). Development of generalized free surface flow models using finite element techniques. In *Proceedings of the Second International Conference on Finite Elements in Water Resources*. Pentech Press.

General Accounting Office (1980). *Costly wastewater treatment plants fail to perform as expected*. Report (CED-81-9) to House Committee on Public Works and Transportation.

Geselbracht, J. J., and Johnston, D. M. (1988). Issues in rule base development. *J. of Water Resources Planning and Management*, 114(4), 457-468.

Geselbracht, J. J., Downey Brill, Jr., E., and Pfeffer, J. T. (1988). Rule-based model of design judgment about sludge bulking. *J. of Environmental Engineering, American Society of Civil Engineers (ASCE)*, 114(1), 54-73.

Giuse, D. (1990). Efficient knowledge representation systems. *The Knowledge Engineering Review*, 5(1), 35-50.

- Gould, G. W., and Lees, H. (1960). The isolation and culture of the nitrifying organisms. Part I: *Nitrobacter*. *Can. J. Microbiol.*, 6, 299-307.
- Green, G. P. (1982). An attempt to predict bulking in activated sludge by the development of a method to measure filamentous organisms. In B. Chambers, and E. J. Tomlinson (Eds.), *Bulking of activated sludge: Preventive and remedial methods*. Chichester, England: Ellis Horwood Ltd. 63-72.
- Gruber, T. R. (1989). *The acquisition of strategic knowledge*. Boston, MA: Academic Press, Inc.
- Gujer, W. (1977). Design of a nitrifying activated sludge process with the aid of dynamic simulation. *Prog. Water Tech.*, 9, 323-336.
- Gujer, W., and Henze, M. (1990). Activated sludge modelling and simulation. In P. Grau and the IAWPRC Programme Committee (Eds.), *Proceedings of the fifteenth biennial conference of the International Association on Water Pollution Research and Control, held in Kyoto, Japan 29 July-3 Aug., 1990*. *Water Sci. Tech.*, 23, 1011-1023.
- Hadden, W. J., and Hadden, S. G. (1985). *Expert systems for environmental regulation*. In K. N. Karna (Ed.), *Expert systems in government symposium*. IEEE Computer Society. 558-566.
- Haldane, J. B. S. (1965). *Enzymes*. Cambridge, MA: MIT Press.
- Hall, E. R., and Murphy, K. L. (1980). Estimation of nitrifying biomass and kinetics in wastewater. *Water Research*, 14, 279-304.
- Hall, E. R., and Murphy, K. L. (1985). Sludge age and substrate effects on nitrification kinetics. *J. Water Pollut. Control Fed.*, 57(5), 413-418.

- Hanaki, K., Wantawin, C., and Ohgaki, S. (1990). Nitrification at low levels of dissolved oxygen with and without organic loading in a suspended-growth reactor. *Water Research*, 24(3), 297-302.
- Hao, O. J., and Neethling, J. B. (1987). Effect of ratio correlation on data interpretation. *J. of Environmental Engineering, American Society of Civil Engineers (ASCE)*, 113(1), 205-211.
- Harmon, P., and King, D. (1985). *Expert systems: Artificial intelligence in business*. New York, NY: John Wiley & Sons, Inc.
- Henze, M. (Ed.) (1986). *Modelling of biological wastewater treatment. Water Sci. Tech.*, 18(6). Oxford, England: Pergamon Press.
- Henze, M., Grady, Jr., C. P. L., Gujer, W., Marais, G. v. R., and Matsuo, T. (1987a). A general model for single-sludge wastewater treatment systems. *Water Research*, 21(5), 505-515.
- Henze, M., Grady, Jr., C. P. L., Gujer, W., Marais, G. v. R., and Matsuo, T. (1987b). *Final Report: Activated sludge process. IAWPRC Scientific and Technical Reports, No. 1*. London.
- Herrod, R. A., and Tietz, L. (1989). On-line expert systems and their role in the control systems of the future. In *Advances in Instrumentation*. Research Triangle Park, NC: ISA. 449-456.
- Hoffman, T., and Lees, H. (1953). The biochemistry of nitrifying organisms IV. The respiration and intermediate metabolism of *Nitrosomonas*. *Biochem. J.*, 54, 575-583.
- Hollocher, T. C., Tate, M. E., and Nicholas, D. J. (1981). Oxidation of ammonia by *Nitrosomonas europaea*: Definite ¹⁸O tracer evidence that hydroxylamine formation involves a monooxygenase. *Biochem. J.*, 256(21), 10834-10836.

- Horan, N. J., and Eccles, C. R. (1986). The potential for expert systems in the operation and control of activated sludge plants. *Process Biochemistry PRBCAP*, 21(3), 81-85.
- Houck, M. H. (1985). Designing an expert system for real-time reservoir system operation. *Civil Engineering Systems*, 2, 30-37.
- Huang, M. S., Sheno, S., Mathews, A. P., Lai, F.S., and Fan, L. T. (1986). Fault diagnosis of hazardous waste incineration facilities using a fuzzy expert system. In C. N. Kostem, and M. L. Maher (Eds.), *Expert systems in civil engineering*. New York, NY: American Society of Civil Engineers. 145-158.
- Hushon, J. M. (1986). Response to chemical emergencies. *Env. Sci. and Tech.*, 20(2), 118-121.
- James, M. L., Smith, G. M., and Wolford, J. C. (1985). *Applied numerical methods for digital computation*, (3rd Ed.). New York, NY: Harper & Row, Publishers.
- Jayawardena, A. W., and Lai, F. (1989). Time series analysis of water quality data in Pearl River, China. *J. of Environmental Engineering, American Society of Civil Engineers (ASCE)*, 115(3), 590-607.
- Jenkins, D., Neethling, J. B., Bode, H., and Richard, M. G. (1982). The use of chlorination for control of activated sludge bulking. In B. Chambers, and E. J. Tomlinson (Eds.), *Bulking of activated sludge: Preventive and remedial methods*. Chichester, England: Ellis Horwood Ltd. 187-205.
- Jenkins, W. O., and Jowitt, P. W. (1987). Expert systems in river basin management. *Civil Engineering Systems*, 4, 31-38.
- Johnston, D. M. (1985). Diagnosis of wastewater treatment processes. In *Proceedings of the Specialty Conference, Computer Applications*

in Water Resources, Buffalo, New York. New York, NY: American Society of Civil Engineers. 601-610.

Jorgensen, S. E., and Gromiec, M. J. (Eds.) (1985). *Mathematical models in biological waste water treatment.* Amsterdam, the Netherlands: Elsevier Science Publishers B. V.

Junkins, R., Deeny, K., and Eckhoff, T. (1983). *The activated sludge process: Fundamentals of operation.* Ann Arbor.

Kalish, D., Montague, R., and Mar, G. (1980). *Logic: Techniques of formal reasoning, (2nd Ed.).* San Diego, CA: Harcourt-Brace-Jovanovich Publishers.

Kayser, R. (1990). Process control and expert systems for advanced wastewater treatment plants. In R. Briggs (Ed.), *Instrumentation, control and automation of water and wastewater treatment and transport systems. Proceedings of the 5th IAWPRC Workshop held in Yokohama and Kyoto, Japan 26 July-3 Aug., 1990.* Oxford, England: Pergamon Press. 203-210.

Kerri, K. D. (1991). Measurement of operator training benefits. In P. Grau and the IAWPRC Programme Committee (Eds.), *Proceedings of the fifteenth biennial conference of the International Association on Water Pollution Research and Control, held in Kyoto, Japan 29 July-3 Aug., 1990.* *Water Sci. Tech.*, 23, 1811-1816.

Knight, B. (1987). An expert system for upgrading small water supplies. *Civil Engineering Systems*, 4, 27-30.

Koskinen, K. (1989). Expert systems as a top level controller for activated sludge process. *Water Sci. Tech.*, 21, 1809-1812.

Koskinen, K., and Viitasaari, M. (1990). Rule-based expert system for the control of the activated sludge process. In R. Briggs (Ed.), *Instrumentation, control and automation of water and wastewater*

treatment and transport systems. Proceedings of the 5th IAWPRC Workshop held in Yokohama and Kyoto, Japan 26 July-3 Aug., 1990. Oxford, England: Pergamon Press. 243-249.

Kraiem, Z. M., Dagli, D. P., and Diekmann, J. E. (1989). DISCON: An expert system for the analysis of differing site conditions claims. *Knowledge-Based Systems J.*, 2(3), 158-163.

Krichten, D. J., Wilson, K. D., and Tracy, K. D. (1990). Application of expert system technology to control of biological phosphorus removal plant. In R. Briggs (Ed.), *Instrumentation, control and automation of water and wastewater treatment and transport systems. Proceedings of the 5th IAWPRC Workshop held in Yokohama and Kyoto, Japan 26 July-3 Aug., 1990.* Oxford, England: Pergamon Press. 195-201.

Kunz, J., Shortliffe, E. H., Buchanan, B. G., and Feigenbaum, E. A. (1983). *A comparison of techniques of computer-assisted decision making in medicine.* Report No. HPP-83-32, Heuristic Programming Project, Stanford University.

Kyburg, Jr., H. E. (1987). Bayesian and non-bayesian evidential updating. *Artificial Intelligence*, 31, 271-293.

Lai, W., and Berthouex, P. M. (1990). Testing expert system for activated sludge process control. *J. of Environmental Engineering, American Society of Civil Engineers (ASCE)*, 116(5), 890-909.

Laudelot, H., and Van Tichelen, L. (1960). Kinetics of the nitrite oxidation by *Nitrobacter winogradskyi*. *J. Bact.*, 79, 39-42.

Law, K. H., Zimmie, T. F., and Chapman, D. R. (1986). An expert system for in active hazardous waste site characterization. In C. N. Kostem, and M. L. Maher (Eds.), *Expert systems in civil engineering.* New York, NY: American Society of Civil Engineers. 159-173.

- Laws, W. (1991). Keys to successful implementation of computer applications. *Cal. Water Pollut. Control Assoc. Bulletin*, 28(3), 20-22.
- Lees, H. (1954). Biochemistry of the nitrifying bacteria. In B. A. Fry and J. L. Peel (Eds.), *Autotrophic microorganisms*. London: Cambridge University Press. 84-98.
- Levitt, R. E. (1986). HOWSAFE: A microcomputer-based expert system to evaluate the safety of a construction firm. In C. N. Kostem and M. L. Maher (Eds.), *Expert systems in civil engineering*. New York, NY: American Society of Civil Engineers.
- Levitt, R. E., and Kartam, N. A. (1990). Expert systems in construction engineering and management: State-of-the-art. *The Knowledge Engineering Review*, 5(2), 97-125.
- Linkens, D. A. (1990). AI in control systems engineering. *The Knowledge Engineering Review*, 5(1), 181-214.
- Lotter, L. H., and Pitman, A. R. (1990). Biochemical nutrient removal modelling: Activated sludge plant operation. In M. Suzuki, and the JSWPR Programme Committee (Eds.), *Reprint of poster papers, Water Pollution Research and Control*. 489-492.
- Loveless, J. E., and Painter, H. A. (1968). The influence of metal ion concentration and pH value on the growth of a *Nitrosomonas* strain isolated from activated sludge. *J. Gen. Microbiol.*, 52, 1-14.
- Ludvigsen, P. J., Simms, R. C., and Grenney, W. J. (1986). Development of a prototype expert system for assessing organic chemical mobility and degradation to determine soil treatment requirements. In *Fourth Conference on Computing in Civil Engineering*. New York, NY: American Society of Civil Engineers.

- Ludwig, W. H. (1989). Performs other necessary duties as required. *Paper presented at the 62nd Annual Conference of the Water Pollut. Control Fed., San Francisco, CA.*
- Lumbers, J. P., Cook, S. C., and Thomas, G. A. (1989). Modelling of nitrification at the Mogden treatment works. *Water Sci. Tech.*, 21(4-5), 305-314.
- Maeda, K. (1985). An intelligent decision support system for activated sludge wastewater treatment process. In R. A. R. Drake (Ed.), *Advances in Waste Pollution Control: Symposium Instrumentation and Control of Water and Wastewater Treatment and Transport Systems, IAWPRC*. London, England: Pergamon Press. 629-632.
- Maeda, K., Inoue, S., Hirotsuji, J., Nonoyama, M., and Aya, S. (1990). A new expert system based on deep knowledge for water and wastewater treatment plant. In R. Briggs (Ed.), *Instrumentation, control and automation of water and wastewater treatment and transport systems. Proceedings of the 5th IAWPRC Workshop held in Yokohama and Kyoto, Japan 26 July-3 Aug., 1990*. Oxford, England: Pergamon Press. 219-226.
- Manning, A. W., and Dobs, D. M. (1980). *Design handbook for automation of activated sludge wastewater treatment plants*. Cincinnati, OH: U.S. EPA (EPA-600/8-80-028).
- McCorduck, P. (1979). *Machines who think*. San Francisco, CA: W. H. Freeman and Co.
- Medina, Jr., M. A. (1985). Deterministic and statistical water quality modeling. In *Proceedings of the Specialty Conference, Computer Applications in Water Resources, Buffalo, New York*. New York, NY: American Society of Civil Engineers. 1100-1110.

- Meiklejohn, J. (1954). Some aspects of the physiology of the nitrifying bacteria. In B. A. Fry and J. L. Peel (Eds.), *Autotrophic microorganisms*. London: Cambridge University Press. 68-83.
- Metcalf and Eddy, Inc. (1991). *Wastewater engineering: Treatment, disposal, reuse, (3rd Ed.)*. New York, NY: McGraw-Hill Book Co.
- Mikroudis, G. K., Fang, H. Y., and Wilson, J. L. (1986). Development of GEOTOX expert system for assessment of hazardous waste sites. In *First International Symposium on Environmental Geotechnology*. Leigh, PA: Leigh University, Leigh.
- Minsky, M. (1963). *Computers and thought*. New York, NY: McGraw Hill.
- Monod, J. (1949). The growth of bacterial cultures. *Ann. Rev. Microbiol.*, 3, 371-394.
- Nakazato, T., and Okuno, N. (1990). An expert system for inexperienced operators. *Paper presented at the 63rd Annual Conference of the Water Pollut. Control Fed., Washington, D.C.*
- Neethling, J. B. (1983). *The control of activated sludge bulking by chlorination*. Ph.D. Thesis, University of California, Berkeley.
- Neethling, J. B., Jenkins, D., and Johnson, K. M. (1984). Chemistry, microbiology, and modelling of chlorination for activated sludge bulking control. *J. Water Pollut. Control Fed.*, 57(8), 882-889.
- Neopolitan, R. E. (1986). A comparison of the MYCIN model for reasoning under uncertainty to a probability based model. In J. F. Gilmore (Ed.), *Proceedings of SPIE, Application of Artificial Intelligence III*. Orlando, FL: The International Society of Optical Engineers. 375-383.

- Neufeld, R., Greenfield, J., and Rieder, B. (1986). Temperature, cyanide and phenolic nitrification inhibition. *Water Research*, 20(5), 633-642.
- Nolasco, D., Patry, G. G., and Olsson, G. (1989). Knowledge acquisition from simulation models. In *Proceedings of the Sixth Conference on Computing in Civil Engineering: Computers in Engineering Practice, Atlanta, GA, 1989*. New York, NY: American Society of Civil Engineers. 131-138.
- Norusis, M. J., and Jacquez, J. A. (1975). Diagnosis I: Symptom non-independence in mathematical model for diagnosis. *Computers in Biomedical Research*, 8, 156-172.
- Novotny, V., Jones, H., Feng, X., and Capodaglio, A. (1991). Time series analysis models of activated sludge plants. In P. Grau and the IAWPRC Programme Committee (Eds.), *Proceedings of the fifteenth biennial conference of the International Association on Water Pollution Research and Control, held in Kyoto, Japan 29 July-3 Aug., 1990*. *Water Sci. Tech.*, 23, 1107-1116.
- Olsson, G., and Andrews, J. F. (1978). The dissolved oxygen profile--A valuable tool for control of the activated sludge process. *Water Research*, 12, 985-1004.
- Olsson, G., Homberg, U., and Wikstrom, A. (1985). A model library for dynamic simulation of activated sludge systems. In R. A. R. Drake (Ed.), *Advances in Waste Pollution Control: Symposium Instrumentation and Control of Water and Wastewater Treatment and Transport Systems, IAWPRC*. London, England: Pergamon Press. 721-728.
- Olsson, G. (1987). Dynamic modelling, estimation and control of activated sludge systems. In M. B. Beck (Ed.), *Advances in water pollution control: System analysis in water quality management*. Oxford, England: Pergamon Press. 349-356.

- Ostgaard, K., Storen, O., and Indergraad, M. (1985). Microcomputer-assisted recording of environmental variables. *Water Research*, 19(6), 799-803.
- Painter, H. A. (1977). Microbial transformations of inorganic nitrogen. *Prog. Water Tech.*, 8(4/5), 3-29.
- Painter, H. A., and Loveless, J. E. (1983). Effect of temperature and pH value on the growth-rate constants of nitrifying bacteria in the activated-sludge process. *Water Research*, 17(3), 237-248.
- Palm, J., Jenkins, D., and Parker, D. S. (1980). Relationship between organic loading, dissolved oxygen concentration and sludge settleability in the completely-mixed activated sludge process. *J. Water Pollut. Control Fed.*, 52(10), 2484-2506.
- Palmer, R. N. (1985). A review of artificial intelligence. In *Proceedings of the Specialty Conference, Computer Applications in Water Resources, Buffalo, New York*. New York, NY: American Society of Civil Engineers. 591-600.
- Pantea-Kiser, L., Wukasch, R. F., and Alleman, J. E. (1990). The effect of inhibitory compounds on biological nitrification. In *44th Purdue Industrial Waste Conference Proceedings*. Chelsea, Michigan: Lewis Publishers, Inc.
- Parker, D. G., Parker, S. C., and Balagamwala, A. (1989). Linking an environmental expert system with a DBMS. In *Proceedings of the Sixth Conference on Computing in civil engineering: Computers in Engineering Practice, Atlanta, GA, 1989*. New York, NY: American Society of Civil Engineers. 147-153.
- Pasveer, A. (1969). A case of filamentous activated sludge. *J. Water Pollut. Control Fed.*, 41(7), 1340-1352.

- Patry, G. G., and Chapman, D. (Eds.) (1989). *Dynamic modeling and expert systems in wastewater engineering*. Chelsea, MI: Lewis Publishers, Inc.
- Payraudeau, M., Rogalla, F., and Penillard, P. (1990). Temperature influence on complete nitrogen removal with biological aerated filters. In M. Suzuki, and the JSWPR Programme Committee (Eds.), *Reprint of poster papers, Water Pollution Research and Control*. 485-488.
- Pedersen, K. (1989). *Expert systems programming: Practical techniques for rule-based systems*. New York, NY: John Wiley & Sons, Inc.
- Peeters, T. L., Van Gool, A. D., and Laudelout, H. (1969). Kinetics study of oxygen limited respiration in nitrifying bacteria. *Bact. Proc.*, 141.
- Pitman, A. R. (1991). Design considerations for nutrient removal activated sludge plants. In P. Grau and the IAWPRC Programme Committee (Eds.), *Proceedings of the fifteenth biennial conference of the International Association on Water Pollution Research and Control, held in Kyoto, Japan 29 July-3 Aug., 1990*. *Water Sci. Tech.*, 23, 781-790.
- Poduska, R. A. (1973). *A dynamic model of nitrification for the activated sludge process*. Ph.D. dissertation submitted to the faculty of Clemson University.
- Poduska, A., and Andrews, J. F. (1975). Dynamics of nitrification in the activated sludge process. *J. Water Pollut. Control Fed.*, 47(11), 2599-2619.
- Qasim, S. R. (1985). *Wastewater treatment plants: Planning, design, and operation*. New York, NY: Holt, Rinehart and Winston.

- Quillian, M. R. (1968). Semantic memory. In *Semantic information processing*. Cambridge, MA: MIT Press. 227-270.
- Rachwal, A. J., and Johnstone, D. W. M. (1982). The application of settleability tests for the control of activated-sludge plants. In B. Chambers, and E. J. Tomlinson (Eds.), *Bulking of activated sludge: Preventive and remedial methods*. Chichester, England: Ellis Horwood Ltd. 224-242.
- Randall, C. W., and Buth, D. (1984a). Nitrite build-up in activated sludge resulting from temperature effects. *J. Water Pollut. Control Fed.*, 56(9), 1039-1044.
- Randall, C. W., and Buth, D. (1984b). Nitrite build-up in activated sludge resulting from combined temperature and toxicity effects. *J. Water Pollut. Control Fed.*, 56(9), 1045-1049.
- Reboh, R., Reiter, J., and Gaschnig, J. (1982). *Development of a knowledge-based interface to a hydrological simulation program*. Menlo Park, CA: SRI International.
- Rensink, J. H. (1974). New approach to preventing bulking sludge. *J. Water Pollut. Control Fed.*, 46(8), 1888-1894.
- Rensink, J. H., Donker, H. J., and Ijwema, T. S. J. (1982). The influence of feed pattern on sludge bulking. In B. Chambers, and E. J. Tomlinson (Eds.), *Bulking of activated sludge: Preventive and remedial methods*. Chichester, England: Ellis Horwood Ltd. 147-163.
- Richard, M., Hao, O., and Jenkins, D. (1985). Growth kinetics of *Sphaerotilus* species and their significance in activated sludge bulking. *J. Water Pollut. Control Fed.*, 57(1), 68-81.
- Richard, M. (1989). *Activated sludge microbiology*. Alexandria, VA: Water Pollut. Control Fed.

- Richardson, M. (1985). *Nitrification, inhibition in the treatment of sewage*. Reading, UK: Nugent House.
- Roesner, L. A., Gigure, P. R., and Evenson, D. E. (1981). *Computer program documentation for the stream water quality model QUAL-II*. Athens, GA: U.S. EPA (EPA-600/9-81-014).
- Rosenblatt, F. (1957). *The PERCEPTRON: A perceiving and recognizing automation*. New York, NY: Cornell Aeronautical Lab.
- Rossman, L. A., and Siller, T. J. (1987). Expert systems in environmental engineering. In M. L. Maher (Ed.), *Expert systems for civil engineers: Technology and application*. New York, NY: American Society of Civil Engineers.
- Rossman, L. A. (1989). A hybrid knowledge-based/algorithmic approach to the design of waste treatment systems. In *Proceedings of the Sixth Conference on Computing in Civil Engineering: Computers in Engineering Practice, Atlanta, GA, 1989*. New York, NY: American Society of Civil Engineers. 162-169.
- Roussel, P. (1975). *Manual d'utilisation. Rapporte Interne GIA, Ver de Luminy, Univ. d'Aix Marseille*. Luminy, France.
- Rowan, D. A. (1987). Using an expert system for fault diagnosis. *Control Engineering*, 34(9), 160-164.
- Rozich, A. F., and Castens, D. J. (1986). Inhibition kinetics of nitrification in continuous-flow reactors. *J. Water Pollut. Control Fed.*, 58(3), 220-226.
- Sacks, P. H., Patterson, A. M., and Tunes, M. H. M. (1986). ESCORT: An expert system for complex operations in real-time. *Expert Systems*, 3(1).

- Schroeder, E. D. (1977). *Water and wastewater treatment*. New York, NY: McGraw-Hill Book Co.
- Shieh, W. K., and Mulcahy, L. T. (1986). Experimental determination of intrinsic kinetic coefficients for biological wastewater treatment systems. *Water Sci. Tech.*, 18, 1-10.
- Shortliffe, E. H. (1976). *Computer-based medical consultation*. New York, NY: American Elsevier.
- Shortliffe, E. H., Buchanan, B. G., and Feigenbaum, E. A. (1979). Knowledge engineering for medical decision making: A review of computer-based clinical decision aids. In *Proceedings of the IEEE* 67(9), 1207-1224.
- Siddall, J. N. (1984). *The application of artificial intelligence to engineering design*. Hamilton, Ontario: McMaster University.
- Skinner, F. A., and Walker, N. (1961). Growth of *Nitrosomonas europaea* in water and continuous culture. *Archs. Mikrobiol.*, 38, 339-349.
- Slijkhuis, H., and Deinema, M. H. (1982). The physiology of *Microthix parvicella*, a filamentous bacterium isolated from activated sludge. In B. Chambers, and E. J. Tomlinson (Eds.), *Bulking of activated sludge: Preventive and remedial methods*. Chichester, England: Ellis Horwood Ltd. 75-83.
- Sollfrank, U., and Gujer, W. (1991). Characterization of domestic wastewater for mathematical modelling of the activated sludge process. In P. Grau and the IAWPRC Programme Committee (Eds.), *Proceedings of the fifteenth biennial conference of the International Association on Water Pollution Research and Control, held in Kyoto, Japan 29 July-3 Aug., 1990*. *Water Sci. Tech.*, 23, 1057-1066.

- Spiegel, M. R. (1961). *Statistics*. New York, NY: McGraw-Hill Book Co.
- Stafford, D. A. (1974). The effect of phenols and heterocyclic bases in nitrification in activated sludges. *J. Appl. Bact.*, 37, 75-82.
- Stainer, R. Y., Ingraham, J. L., Wheelis, M. L., and Painter, P. R. (1986). *The microbial world, (5th Ed.)*. Englewood Cliffs, NJ: Prentice-Hall.
- Stenstrom, M. K., and Andrews, J. F. (1979). Real-time control of activated sludge process. *J. of the Environmental Engineering Division, ASCE*, 105(EF2), 245-260.
- Stenstrom, M. K., and Ng, A. S. (1984). *Powdered activated carbon enhanced nitrification of petroleum wastewater treatment*. Technical Report (UCLA-ENG-84-11).
- Stenstrom, M. K., and Poduska, R. A. (1980). The effect of dissolved oxygen concentration on nitrification. *Water Research*, 14, 643-649.
- Stenstrom, M. K., Kido, W., Shanks, R. F., and Mulkerin, M. (1989). Estimating oxygen transfer capacity of a full-scale pure oxygen activated sludge plant. *J. Water Pollut. Control Fed.*, 61(2), 208-220.
- Stenstrom, M. K., and Song, S. S. (1991). Effects of oxygen transport limitation on nitrification in the activated sludge process. *Res. J. Water Pollut. Control Fed.*, 63(3), 208-219.
- Stevens, D. K., Berthouex, P. M., and Chapman, T. W. (1989). Dynamic model of nitrification in fluidized bed. *J. of Environmental Engineering, American Society of Civil Engineers (ASCE)*, 115(5), 910-929.

- Sujarittanonta, S., and Sherrard, J. H. (1981). Activated sludge nickel toxicity studies. *J. Water Pollut. Control Fed.*, 53(8), 1314-1322.
- Sykes, J. F., Soyupak, S., and Farquhar, G. J. (1982). Modelling of leachate organic migration and attenuation in groundwaters below sanitary landfills. *Water Resources Research*, 18(1), 135-145.
- Tang, C. C., Brill, Jr., D. E., and Pfeffer, J. T. (1987). Comprehensive model of activated sludge wastewater treatment system. *J. of Environmental Engineering, American Society of Civil Engineers (ASCE)*, 113(5), 952-969.
- Tang, D. H., and Pinder, G. F. (1977). Simulation of groundwater flow and mass transport under uncertainty. *Advances in Water Resources*, 1, 25-30.
- Tchobanoglous, G., and Schroeder, E. D. (1985). *Water quality: Characteristics, modeling, modification*. Reading, MA: Addison-Wesley Publishing Co.
- Tenney, M. W., and Stumm, W. (1965). Chemical flocculation of microorganisms in biological waste treatment. *J. Water Pollut. Control Fed.*, 37(10), 1370-1388.
- Texas Instruments, Inc. (1988). *Personal Consultant Plus*. Austin, TX.
- Tomei, M. C., Di Pinto, A. C., and Ramadori, R. (1990). ASCAM: A simplified model for biological nitrogen removal. In M. Suzuki, and the JSWPR Programme Committee (Eds.), *Reprint of poster papers, Water Pollution Research and Control*. 449-452.
- Tomlinson, E. J. (1982). The emergence of the bulking problem and the current situation in the UK. In B. Chambers, and E. J. Tomlinson (Eds.), *Bulking of activated sludge: Preventive and remedial methods*. Chichester, England: Ellis Horwood Ltd. 17-23.

- Tomlinson, T. G., Boon, A. G., and Trotman, C. W. A. (1966). Inhibition of nitrification in the activated sludge process of sewage disposal. *J. Appl. Bact.*, 29, 266-291.
- Vaccari, D. A., and Christodoulatos, C. (1990). Estimation of the Monod model coefficients for dynamic systems using actual activated sludge plant data. In R. Briggs (Ed.), *Instrumentation, control and automation of water and wastewater treatment and transport systems. Proceedings of the 5th IAWPRC Workshop held in Yokohama and Kyoto, Japan 26 July-3 Aug., 1990*. Oxford, England: Pergamon Press. 449-454.
- Van den Eynde, E., Houtmeyers, J., and Verachtert, H. (1982). Relation between substrate feeding pattern and development of filamentous bacteria in activated sludge. In B. Chambers, and E. J. Tomlinson (Eds.), *Bulking of activated sludge: Preventive and remedial methods*. Chichester, England: Ellis Horwood Ltd. 128-142.
- Van Droogenbroeck, R., and Laudelout, H. (1967). Phosphate requirements of the nitrifying bacteria. *Antonie van Leeuwenhoek*, 33, 287-296.
- Van Haandel, A. C., Ekama, G. A., and Marais, G. v. R. (1981). The activated sludge process--3: Single sludge denitrification. *Water Research*, 15, 1135-1152.
- Van Melle, W. (1979). A domain-dependent production rule system for consultation programs. In *Proceedings of the Sixth IJCAI 79 Joint Conference on Artificial Intelligence*. Tokyo, Japan. 923-925.
- Van Niekerk, A. M., Jenkins, D., and Richard, M. G. (1988). A mathematical model of the carbon-limited growth of filamentous and floc-forming organisms in low F/M sludge. *J. Water Pollut. Control Fed.*, 60(1), 100-106.

- Vargues, H., and Brisou, J. (1961). Researches in nitrifying bacteria in ocean depths on the coast of Algeria. In C. H. Oppenheimer (Ed.), *Symposium on Marine Microbiology*. Springfield, Illinois.
- Vernick, A. S., Lanik, P. D., Langer, B. S., and Hruday, S. E. (1984). The management of wastewater from the petroleum refining industry. In D. Barnes, C. F. Forster, and S. E. Hruday (Eds.), *Surveys in industrial wastewater treatment, vol. 2: Petroleum and organic chemical industries*. Boston, MA: Pitman Advanced Publishing Program.
- Verstraete, W., and Alexander, M. (1972). Heterotrophic nitrification by *Arthrobacter* sp. *J. Bact.*, 110, 955-961.
- Vitasovic, Z., and Andrews, J. F. (1987). A rule-based control system for the activated sludge process. In M. B. Beck (Ed.), *Advances in water pollution control: System analysis in water quality management*. Oxford, England: Pergamon Press. 423-432.
- Wagner, F. (1982). Study of the causes and prevention of the sludge bulking in Germany. In B. Chambers, and E. J. Tomlinson (Eds.), *Bulking of activated sludge: Preventive and remedial methods*. Chichester, England: Ellis Horwood Ltd. 29-40.
- Waller, C. B., and Hurley, B. J. E. (1982). Some experiences in the control of bulking activated sludge. In B. Chambers, and E. J. Tomlinson (Eds.), *Bulking of activated sludge: Preventive and remedial methods*. Chichester, England: Ellis Horwood Ltd. 211-220.
- Wang, S. Y., and Chatterji, G. B. (1984). A finite element model for simulating groundwater flow in a multi-layer aquifer. In *Proceedings of 48th Annual Meeting*. Biloxi, MS: Mississippi Academy of Sciences.

- Warwick, J. J., and Shetty, P. (1988). In-stream nitrification rate prediction. *Water Research*, 22(6), 723-732.
- Watanabe, Y., Lee, C., Koike, M., and Ishiguro, M. (1990). Nitrification kinetics and simultaneous removal of biomass and phosphorus in rotating biological contactors. *Water Sci. Tech.*, 22(3/4), 169-178.
- Wentzel, M. C., Ekama, G. A., and Marais, G. v. R. (1991). Kinetics of nitrification denitrification biological excess phosphorus removal systems--a review. In P. Grau and the IAWPRC Programme Committee (Eds.), *Proceedings of the fifteenth biennial conference of the International Association on Water Pollution Research and Control, held in Kyoto, Japan 29 July-3 Aug., 1990*. *Water Sci. Tech.* , 23, 555-565.
- Wigan, M. R. (1986). Engineering tools for building knowledge-based systems on microsystems. *Microcomputers in Civil Engineering*, 1, 52-68.
- Wild, Jr., Sawyer, C. N., and McMahon, T. C. (1971). Factors affecting nitrification kinetics. *J. Water Pollut. Control Fed.*, 43(9), 1845-1854.
- Willey, R. G. (1982). River and reservoir systems water quality modeling capability. *Technical Paper No. 83*. Hydrologic Engineering Center.
- Wilson, T. E. (1983). Application of ISV test to the operation of activated sludge plants. *Water Research*, 17(6), 707-714.
- WPCF (1977). *Wastewater treatment plant design. Manual of practice No. 8*. Washington D. C.: Water Pollut. Control Fed.

- WPCF (1980). *Wastewater sampling for process and quality control. Manual of practice No. OM-1.* Washington D. C.: Water Pollut. Control Fed.
- WPCF (1985). *Clarifier design. Manual of practice No. FD-8.* Washington D. C.: Water Pollut. Control Fed.
- WPCF (1987). *Activated sludge. Manual of practice No. OM-9.* Alexandria, VA: Water Pollut. Control Fed.
- WPCF (1989). *PROBE: Operating activated sludge using oxygen uptake. Training module No. 1.* Alexandria, VA: Water Pollut. Control Fed.
- WPCF (1990). *Operation of wastewater treatment plants. Manual of practice No. 11 (2nd Ed.).* Alexandria, VA: Water Pollut. Control Fed.
- Yang, X. (1989). *Expert system reasoning under uncertainty with applications to incineration systems.* Ph.D. dissertation submitted to University of California, Los Angeles.
- Yoshioka, T., Teraj, H., and Saiyo, Y. (1982). Growth kinetics studies of nitrifying bacteria by the immunofluorescent counting method. *J. General and Appl. Microbiol.*, 28, 169.

Appendix A

ASPX Knowledge Base Logic Trees

LEGENDS

 : Rule number (for rule formulation, see Appendix E)

 : Result code (for description, see Appendix B)

 : Normal path of logic

 : Path of logic for the "In-depth" search

 : Exit from the current logic tree during "Limited" or "Regular" search technique

 : End of the current logic tree

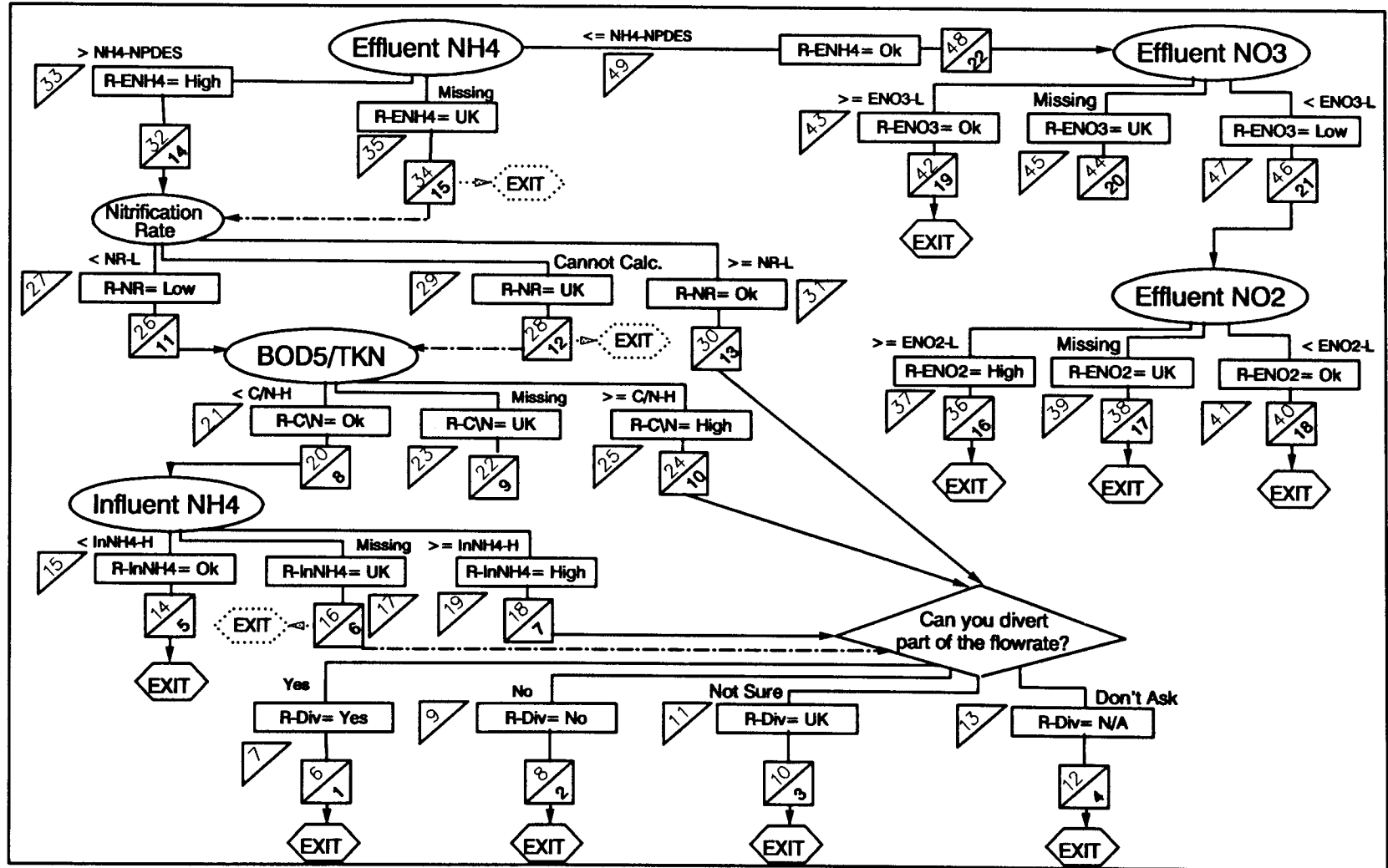


Figure A-1: Complete logic tree for ammonia control

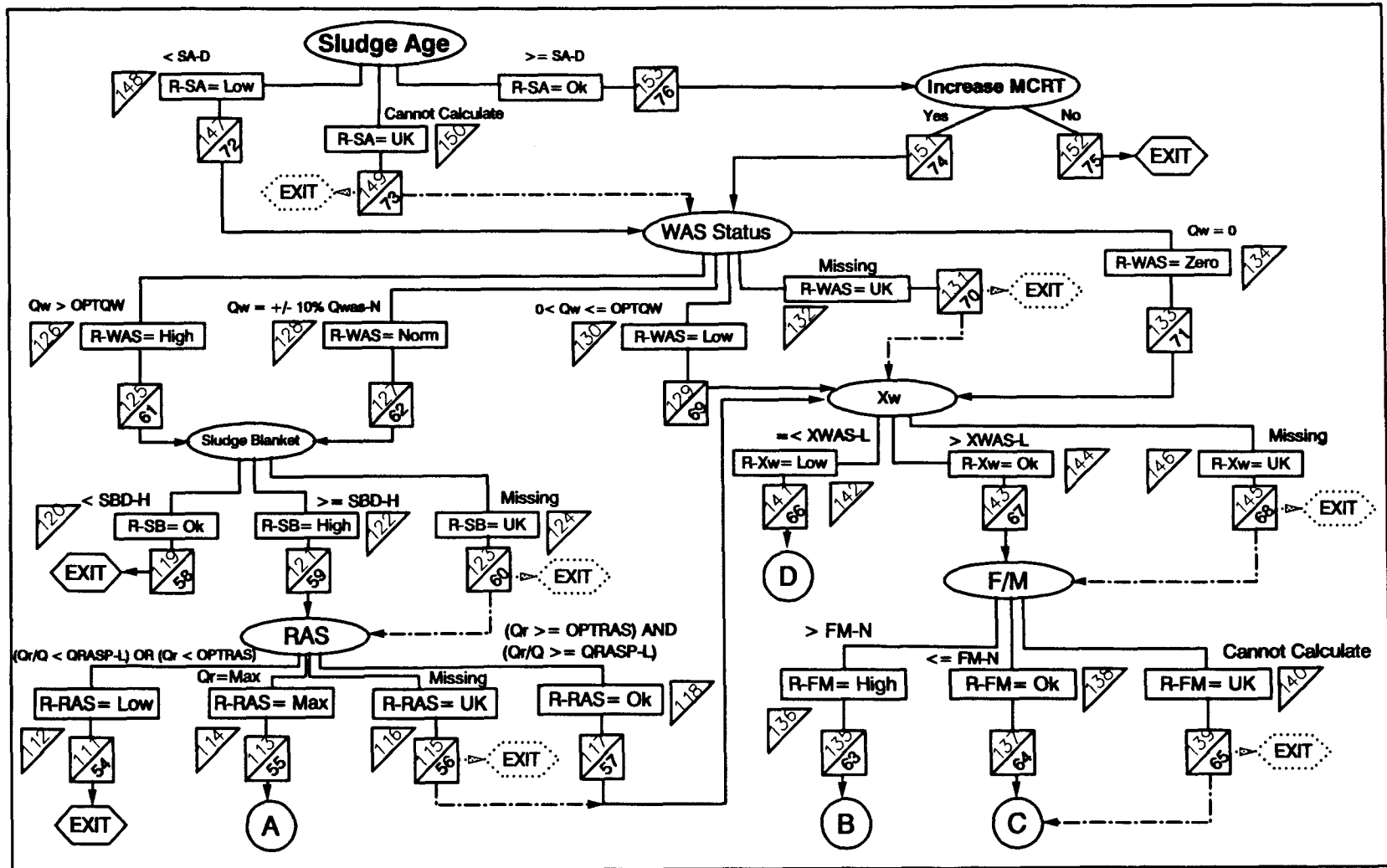


Figure A-2(a): Complete logic tree for the control of sludge age, part 1

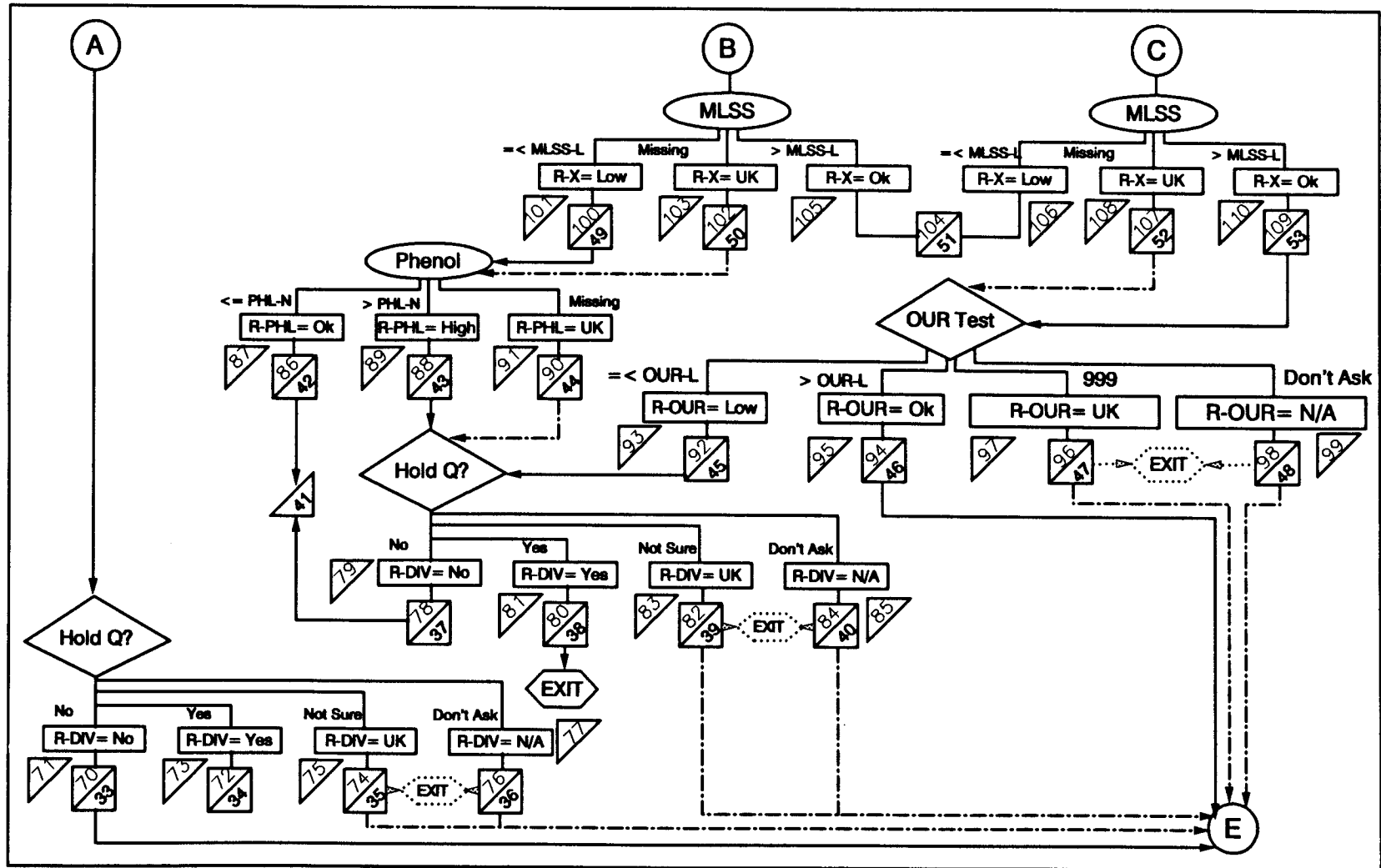


Figure A-2(b): Complete logic tree for the control of sludge age, part 2

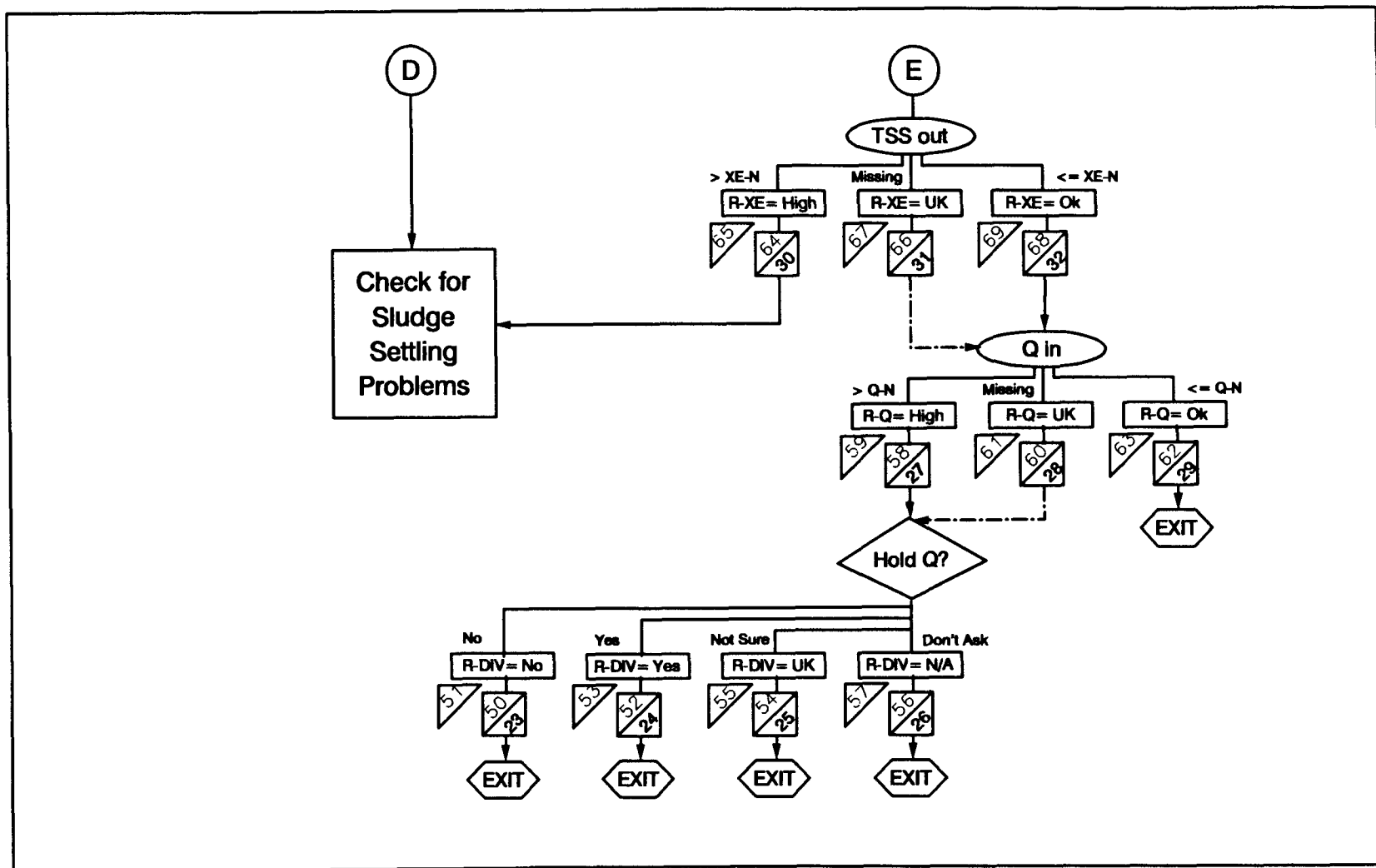


Figure A-2(c): Complete logic tree for the control of sludge age, part 3

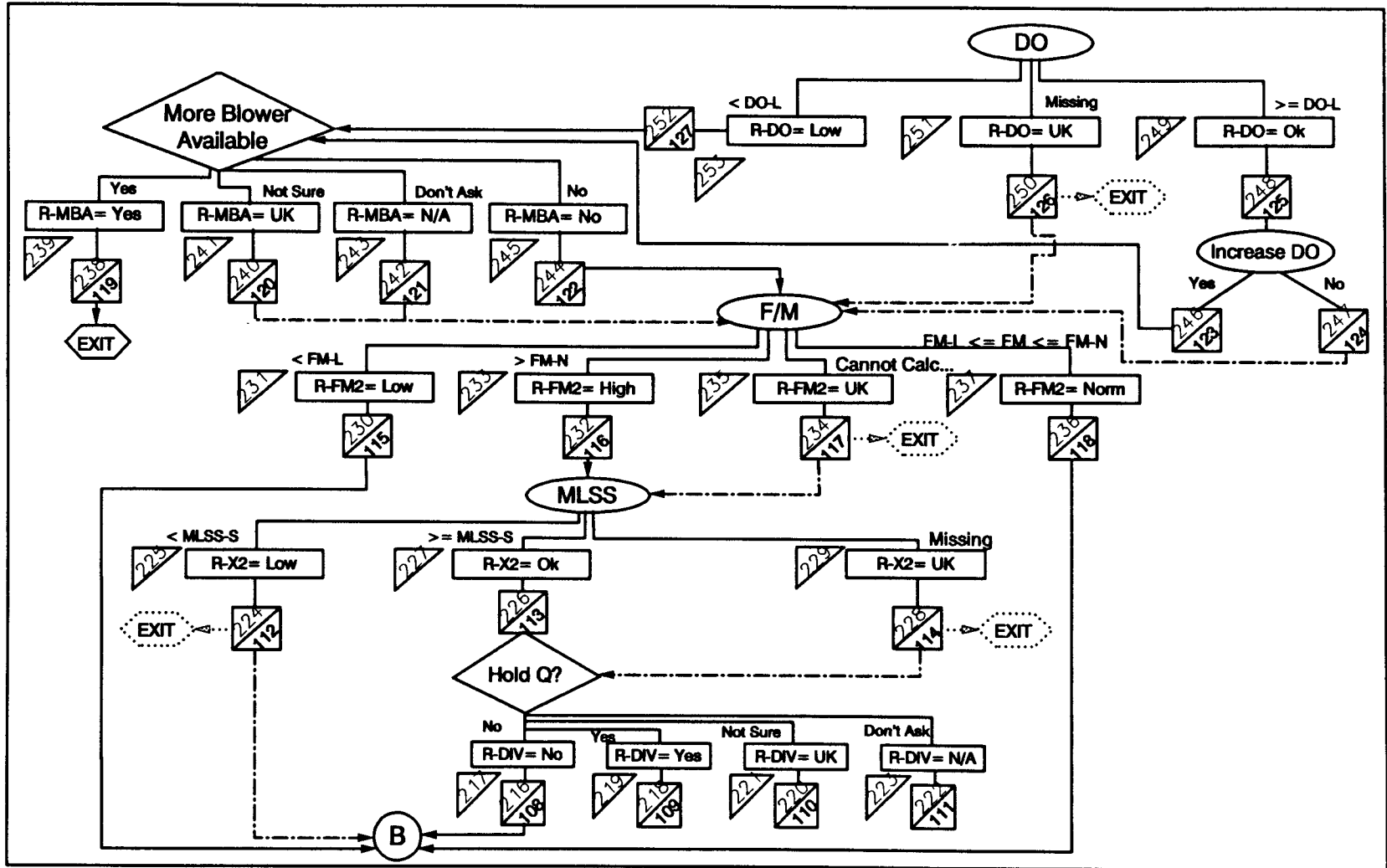


Figure A-3(a): Complete logic tree for the control of DO, part 1

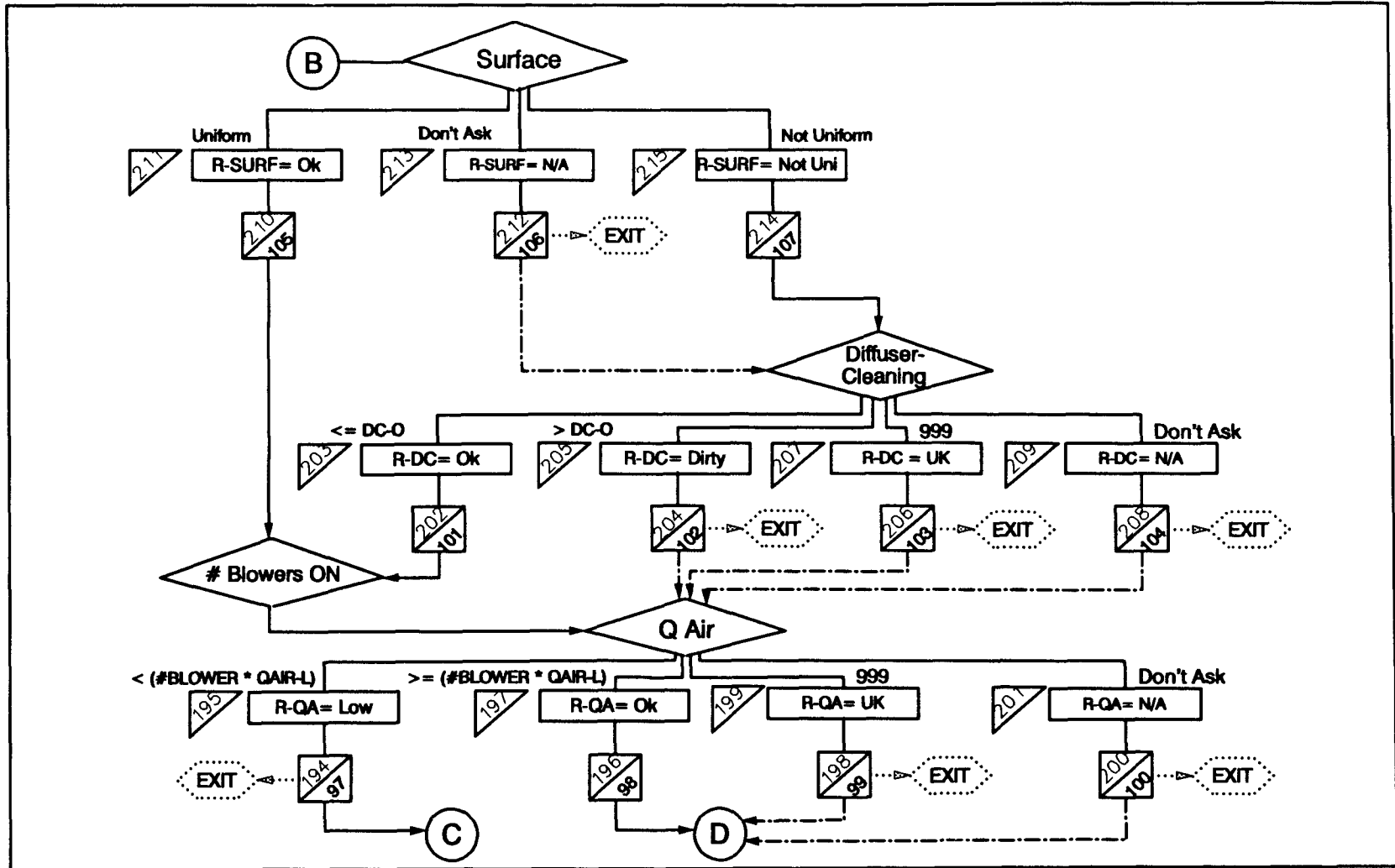


Figure A-3(b): Complete logic tree for the control of DO, part 2

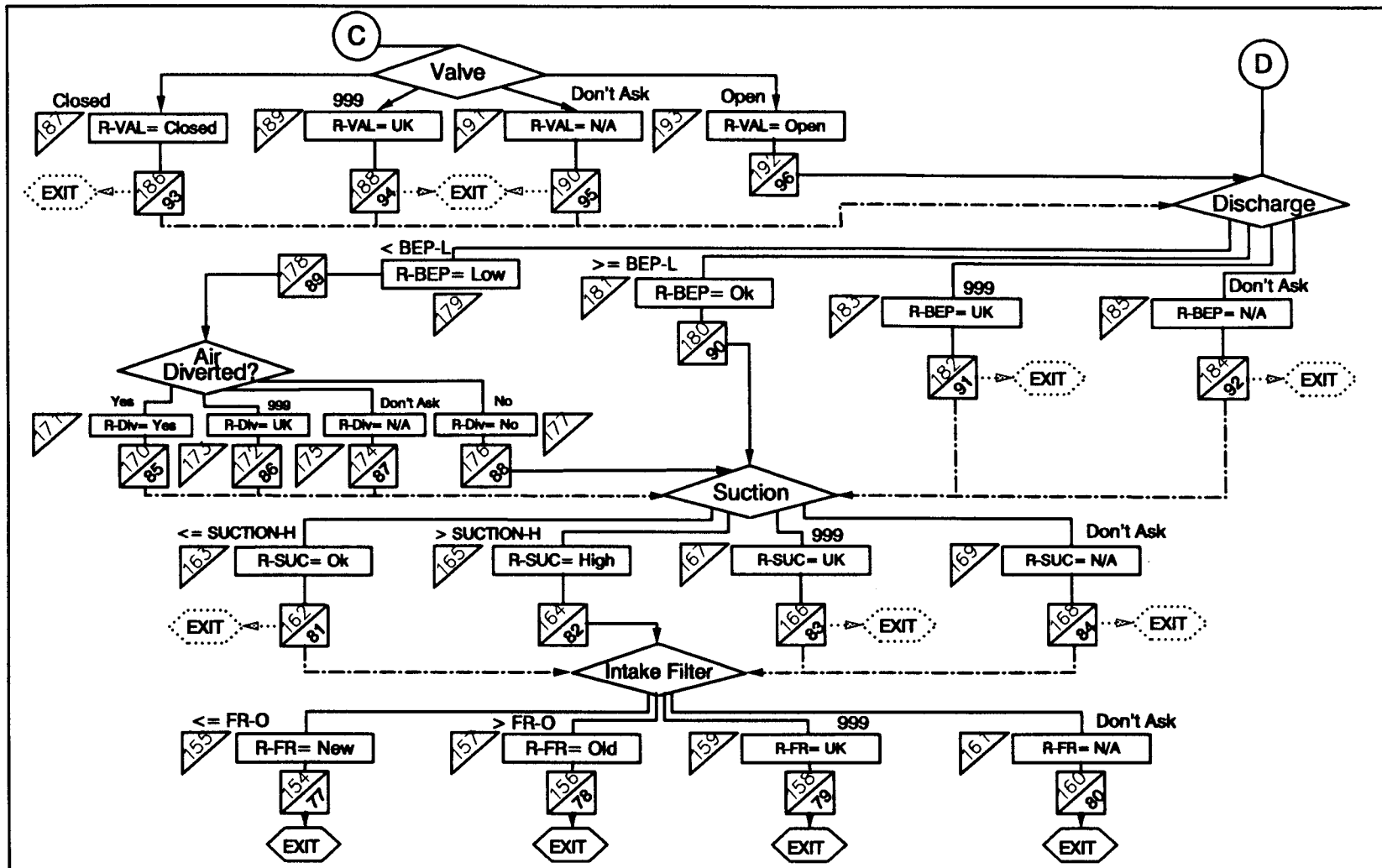


Figure A-3(c): Complete logic tree for the control of DO, part 3

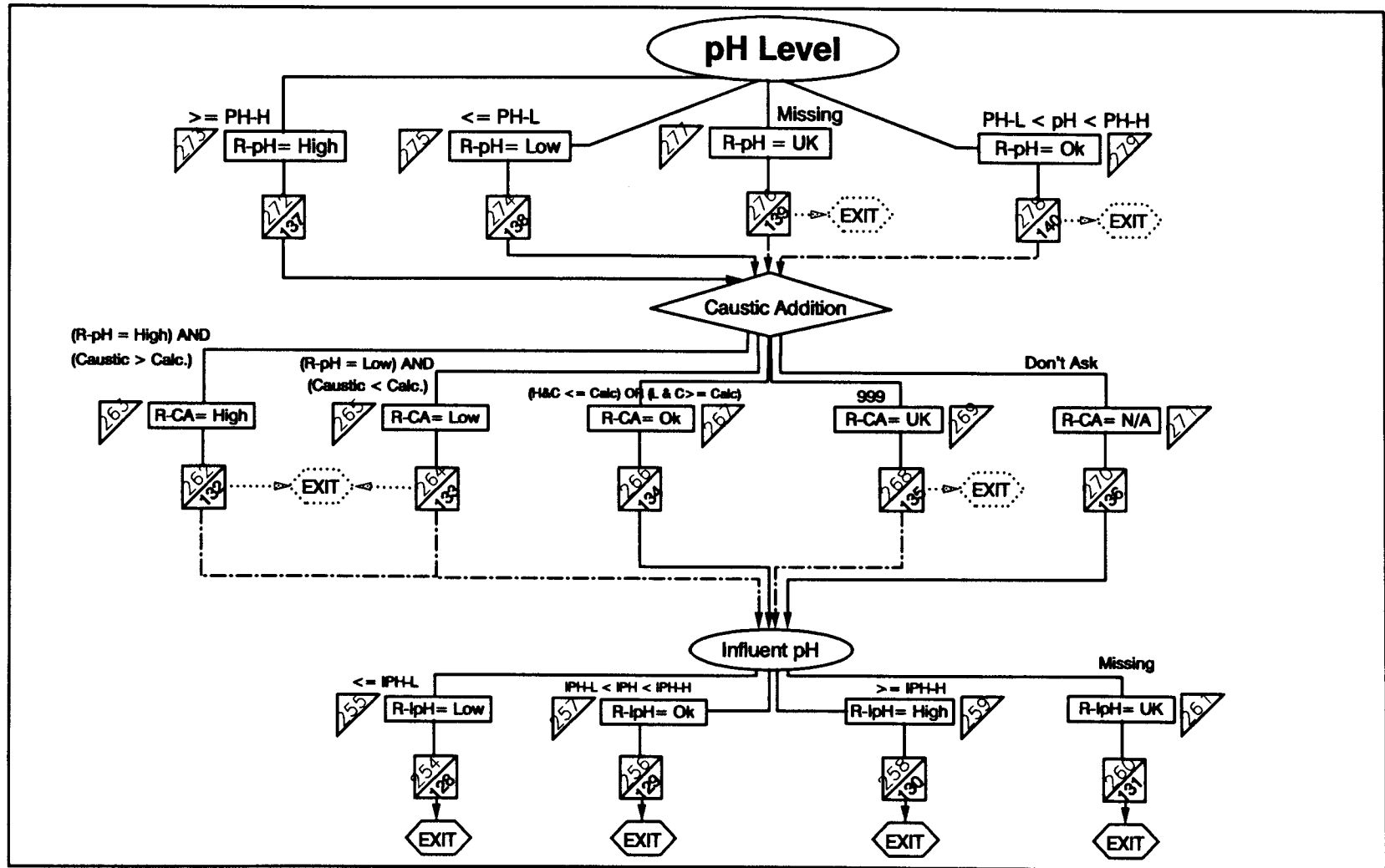


Figure A-4: Complete logic tree for the control of pH concentration

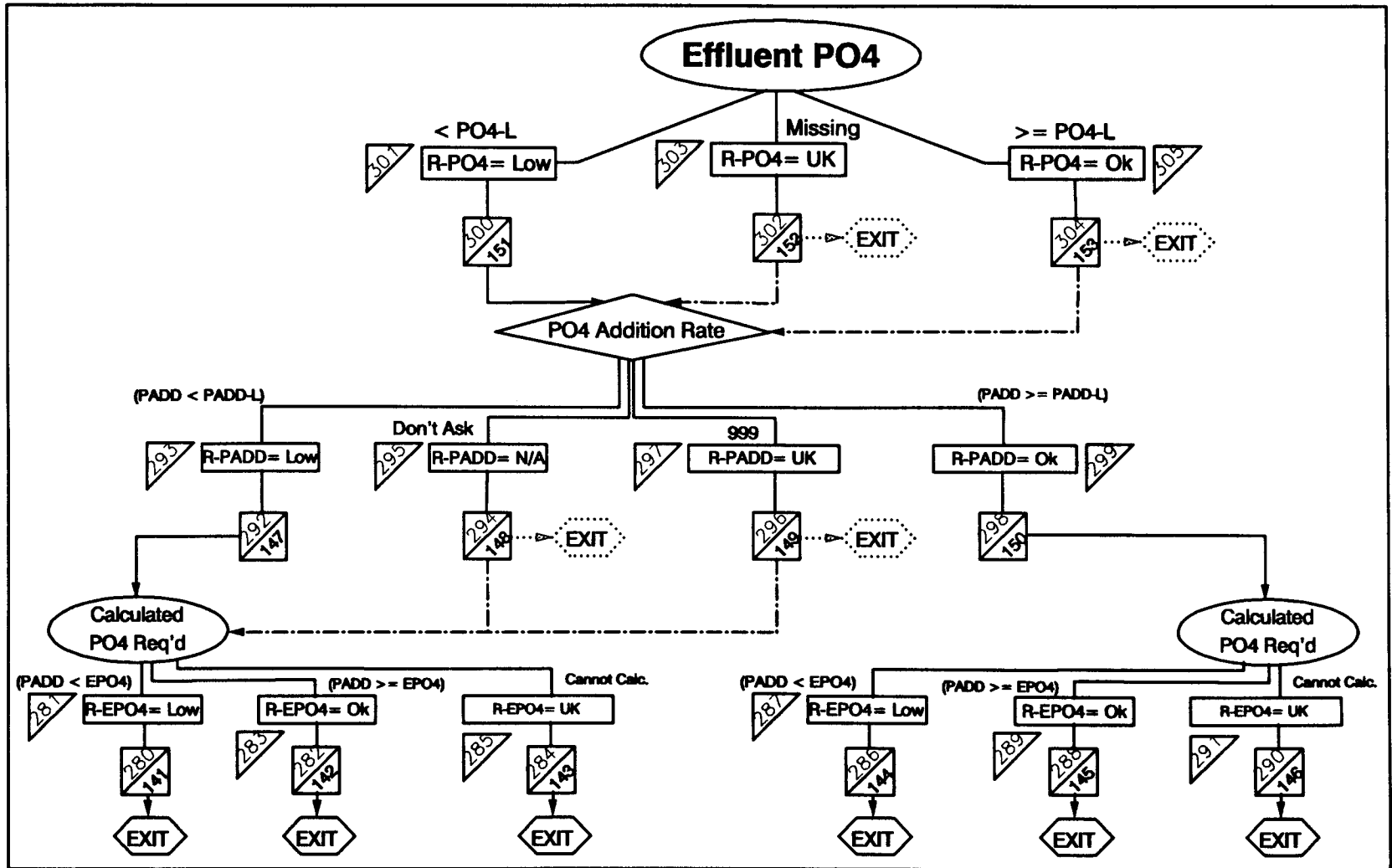


Figure A-5: Complete logic tree for the control of phosphate addition

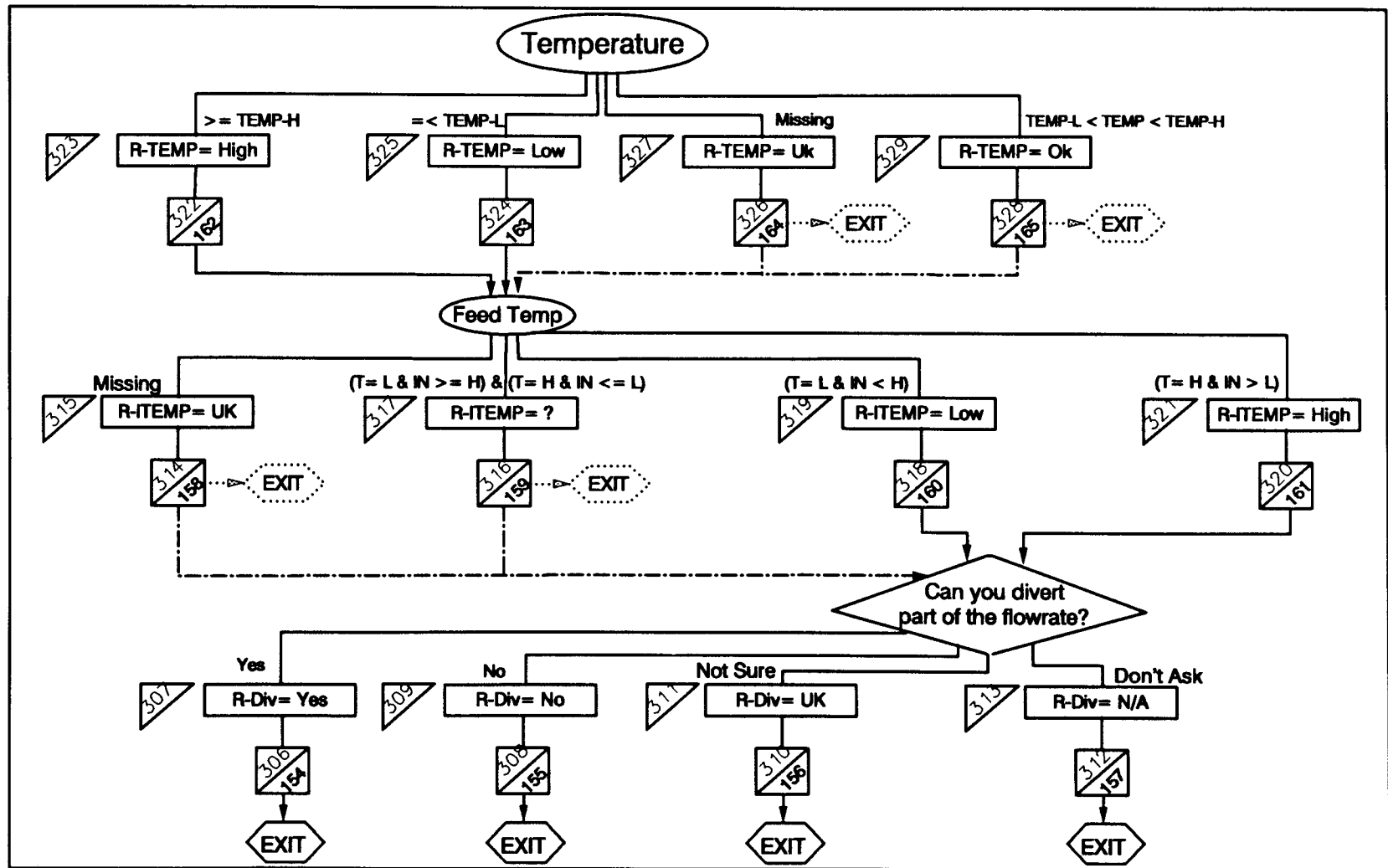


Figure A-6: Complete logic tree for the control of temperature

Appendix B

Result Codes

The result codes here represent the codes used in the logic trees in Appendix A. The corresponding suggestion is displayed based on the nitrification state as follows (where it applies):

- A. Severe Problem
- B. Problem
- C. Potential Problem

X represents a calculated number.

Result Code: 1

Explanation: Yes, we can divert part of the influent flow.

Reliability: CF 100

- A & B & C. Divert X gpm influent flow into temporary holding tanks for X hrs to compensate for high ammonia loading.

Result Code: 2

Explanation: No, we cannot divert part of the influent flow.

Reliability: CF 95

- A & B. Increase DO, sludge age, and MLVSS.
- C. Increase sludge age.

Result Code: 3

Explanation: Not sure if we can divert part of the influent flow.

Reliability: CF 100

- A & B. Consult supervisor to find out if you can divert part of influent flowrate to compensate for high ammonia loading.

Result Code: 4

Explanation: Don't ask whether we can divert part of the influent flow.

Reliability: CF 100

- A. There might be a problem with ammonia concentration, run ASPX again with the "In-depth" option.

- B. There might be a problem with ammonia concentration, run ASPX again with the "Regular" option.

Result Code: 5

Explanation: Influent ammonia is normal.

Reliability: CF 90

- A. Check influent ammonia analysis.

Result Code: 6

Explanation: Influent ammonia concentration is unknown.

Reliability: CF 100

- A & B. a) Measure influent ammonia concentration; b) Update data base; c) Run ASPX again.
- C. a) Measure influent ammonia concentration; b) Update data base.

Result Code: 7

Explanation: Influent ammonia concentration is high.

Reliability: CF 98

- A. a) Inform management about excessive ammonia load; b) find source in the refinery complex and eliminate.
- B & C. Inform management about excessive ammonia load.

Result Code: 8

Explanation: BOD₅/TKN ratio is normal.

Reliability: CF 95

- A. Check BOD₅ and ammonia analysis.

Result Code: 9

Explanation: BOD₅/TKN ratio is unknown.

Reliability: CF 100

- A & B. a) Obtain values for BOD₅ and the influent ammonia concentration; b) Update data base; c) Run ASPX again.
- C. a) Obtain values for BOD₅ and the influent ammonia concentration.

Result Code: 10

Explanation: BOD₅/TKN ratio is high.

Reliability: CF 98

- A. a) Check BOD₅ and ammonia analysis; b) Inform management about unbalanced carbon/nitrogen influent; c) Nitrifier fraction may be low.
- B & C. Nitrifier fraction may be low.

Result Code: 11

Explanation: Nitrification rate is low.

Reliability: CF 80

- A. a) Toxics or inhibitory compounds may be present; b) Check ammonia and MLSS analysis.
- B. Toxics or inhibitory compounds may be present.

Result Code: 12

Explanation: Cannot calculate nitrification rate.

Reliability: CF 100

- A & B. a) Update data base; b) Run ASPX again.

Result Code: 13

Explanation: Nitrification rate is normal.

Reliability: CF 95

- A. Check influent and effluent ammonia and MLVSS analysis.

Result Code: 14

Explanation: Effluent ammonia concentration is high.

Reliability: CF 95

- A & B. Check effluent ammonia analysis.

Result Code: 15

Explanation: Effluent ammonia concentration is unknown.

Reliability: CF 100

- A & B & C. a) Measure effluent ammonia concentration; b) Update data base; c) Run ASPX again.

Result Code: 16

Explanation: Effluent nitrite (NO₂) concentration is high.

Reliability: CF 98

- A & B & C. Partial nitrification is occurring.

Result Code: 17

Explanation: Effluent nitrite (NO₂) concentration is unknown.

Reliability: CF 100

- A. a) Measure effluent nitrite (NO₂) concentration; b) Update data base;
c) Run ASPX again.
- B & C. a) Measure effluent nitrite (NO₂) concentration; b) Update data base.

Result Code: 18

Explanation: Effluent nitrite (NO₂) concentration is normal.

Reliability: CF 95

- A. Check effluent nitrite (NO₂) analysis.

Result Code: 19

Explanation: Effluent nitrate (NO₃) is normal.

Reliability: CF 95

- A. Check effluent nitrate (NO₃) analysis.

Result Code: 20

Explanation: Effluent nitrate (NO₃) is unknown.

Reliability: CF 100

- A & B. a) Measure effluent nitrate (NO₃) concentration; b) Update data base;
c) Run ASPX again.
- C. a) Measure effluent nitrate (NO₃) concentration; b) Update data base.

Result Code: 21

Explanation: Effluent nitrate (NO₃) is low.

Reliability: CF 95

- A. Check effluent nitrate (NO₃) analysis.

Result Code: 22

Explanation: Effluent ammonia concentration is normal.

Reliability: CF 80

- A. a) Check effluent ammonia analysis; b) Check validity of last 3-days data base records.

Result Code: 23

Explanation: No, we cannot divert part of the influent flow.

Reliability: CF 100

- A & B & C. a) Hydraulic overload; b) Search for alternatives to reduce the incoming flowrate.**

Result Code: 24

Explanation: Yes, we can divert part of the influent flow.

Reliability: CF 100

- A & B & C. Divert X gpm influent flow into temporary holding tanks for X hrs to compensate for hydraulic overload.**

Result Code: 25

Explanation: Not sure if we can divert part of the influent flow.

Reliability: CF 100

- A & B. Consult supervisor to find out if you can divert part of influent flowrate to compensate for hydraulic loading.**

Result Code: 26

Explanation: Don't ask whether we can divert part of the influent flow.

Reliability: CF 100

- A. There might be a problem with hydraulic overload, run ASPX again with the "In-depth" option.**
- B. There might be a problem with hydraulic overload, run ASPX again with the "Regular" option.**

Result Code: 27

Explanation: Influent flowrate is high.

Reliability: CF 95

- A. Check influent flowmeter operation.**

Result Code: 28

Explanation: Influent flowrate is unknown.

Reliability: CF 100

- A & B & C. a) Measure influent flowrate; b) Update data base; c) Run ASPX again.**

Result Code: 29

Explanation: Influent flowrate is normal.

Reliability: CF 90

- A. a) Check influent flowmeter operation; b) Undetermined sludge age problem.
- B. Undetermined sludge age problem.

Result Code: 30

Explanation: Effluent suspended solids concentration is high.

Reliability: CF 90

- A. a) Check effluent suspended solids analysis; b) There might be some sludge settlement problems.
- B & C. There might be some sludge settlement problems.

Result Code: 31

Explanation: Effluent suspended solids concentration is unknown.

Reliability: CF 100

- A & B. a) Measure effluent suspended solids; b) Update data base; c) Run ASPX again.
- C. a) Measure effluent suspended solids; b) Update data base.

Result Code: 32

Explanation: Effluent suspended solids concentration is normal.

Reliability: CF 95

- A. Check effluent suspended solids analysis.

Result Code: 33

Explanation: No, we cannot divert part of the influent flow.

Reliability: CF 100

- A & B. Search for alternatives to reduce the incoming flowrate.

Result Code: 34

Explanation: Yes, we can divert part of the influent flow.

Reliability: CF 100

- A & B & C. Divert X gpm influent flow into temporary holding tanks for X hrs to compensate for inadequate RAS rate.

Result Code: 35

Explanation: Not sure if we can divert part of the influent flow.

Reliability: CF 100

- A & B. Consult supervisor to find out if you can divert part of influent flowrate to compensate for inadequate RAS rate.

Result Code: 36

Explanation: Don't ask whether we can divert part of the influent flow.

Reliability: CF 100

- A. There might be a problem with RAS rate, run ASPX again with the "In-depth" option.
- B. There might be a problem with RAS rate, run ASPX again with the "Regular" option.

Result Code: 37

Explanation: No, we cannot divert part of the influent flow.

Reliability: CF 100

- A & B. Search for alternatives to reduce the incoming toxic shocks.

Result Code: 38

Explanation: Yes, we can divert part of the influent flow.

Reliability: CF 100

- A & B & C. Divert X gpm influent flow into temporary holding tanks for X hrs to compensate for high phenol in the influent.

Result Code: 39

Explanation: Not sure if we can divert part of the influent flow.

Reliability: CF 100

- A & B. Consult supervisor to find out if you can divert part of influent flowrate to compensate for high phenol in the influent.

Result Code: 40

Explanation: Don't ask whether we can divert part of the influent flow.

Reliability: CF 100

- A. There might be a problem with high phenol in the influent, run ASPX again with the "In-depth" option.
- B. There might be a problem with high phenol in the influent, run ASPX again with the "Regular" option.

Result Code: 41

Explanation: Increase in RAS rate is requested.

Reliability: CF 90

- A & B. Even though the RAS rate seems normal, increase RAS rate to X gpm to compensate for incoming high organic loading.

Result Code: 42

Explanation: Phenol concentration seems normal.

Reliability: CF 90

- A. Check Phenol analysis.

Result Code: 43

Explanation: Phenol concentration is high.

Reliability: CF 98

- A & B. a) Inform management about high Phenol loading; b) Find and eliminate source in the refinery complex.
C. Inform management about high Phenol loading.

Result Code: 44

Explanation: Phenol concentration is unknown.

Reliability: CF 100

- A & B & C. a) Measure Phenol concentration; b) Update data base; c) Run ASPX again.

Result Code: 45

Explanation: Oxygen Uptake Rate (OUR) test result is low.

Reliability: CF 95

- A & B & C. a) Toxics are likely to be present; b) Inform management about possible toxic spill within the refinery processes; c) Eliminate the source.

Result Code: 46

Explanation: Oxygen Uptake Rate (OUR) test result is normal.

Reliability: CF 95

- A. Confirm the OUR test results.

Result Code: 47

Explanation: Oxygen Uptake Rate (OUR) test result is unknown.

Reliability: CF 100

- A. a) It is highly recommended to perform the OUR test; b) Run ASPX again.
- B. Prepare to perform the OUR test.

Result Code: 48

Explanation: Don't ask the result of the OUR test.

Reliability: CF 100

- A. There might be a problem with toxics, run ASPX again with the "In-depth" option.
- B. There might be a problem with toxics, run ASPX again with the "Regular" option.

Result Code: 49

Explanation: Mixed-Liquor Suspended Solids (MLSS) is low.

Reliability: CF 95

- A. Check MLSS analysis.

Result Code: 50

Explanation: Mixed-Liquor Suspended Solids (MLSS) is unknown.

Reliability: CF 100

- A & B & C. a) Obtain value for MLSS; b) Update data base; c) Run ASPX again.

Result Code: 51

Explanation: Inconsistency in the F/M and MLSS limits.

Reliability: 100

- A & B & C. a) Notify supervisor to revise F/M and MLSS limits; b) Run ASPX again.

Result Code: 52

Explanation: Mixed-Liquor Suspended Solids (MLSS) is unknown.

Reliability: CF 100

- A & B & C. a) Obtain value for MLSS; b) Update data base; c) Run ASPX again.

Result Code: 53

Explanation: Mixed-Liquor Suspended Solids (MLSS) is normal.

Reliability: CF 95

A. Check MLSS analysis.

Result Code: 54

Explanation: Return Activated Sludge (RAS) rate is low.

Reliability: CF 98

A & B & C. a) Increase RAS to X gpm to maintain a proper MLSS; b) Reduce sludge wasting to X gpm.

Result Code: 55

Explanation: Return Activated Sludge (RAS) rate is the highest it can be.

Reliability: CF 95

A & B. Check RAS pump gauges and metering system.

Result Code: 56

Explanation: Return Activated Sludge (RAS) rate is unknown.

Reliability: CF 100

A & B. a) Obtain RAS rate; b) Update data base; c) Run ASPX again.

C. a) Obtain RAS rate; b) Update data base.

Result Code: 57

Explanation: Return Activated Sludge (RAS) rate seems normal.

Reliability: CF 90

A. But increase RAS by X (25%) to maintain a proper sludge blanket depth.

B. But increase RAS by X (10%) to maintain a proper sludge blanket depth.

Result Code: 58

Explanation: Sludge blanket depth is normal.

Reliability: CF 95

A. a) Look for leaks in sludge pipeline; b) Check for malfunctioning sludge level indicator; c) Stop wasting sludge.

- B. a) Check for malfunctioning sludge level indicator; b) Reduce sludge wasting to X gpm.

Result Code: 59

Explanation: Sludge blanket depth is high.

Reliability: CF 90

- A. a) Inform management about potential sludge overflow; b) Look for leaks in sludge pipeline; c) Check for malfunctioning sludge level indicator.
- B. a) Inform management about potential sludge overflow; b) Check for malfunctioning sludge level indicator.

Result Code: 60

Explanation: Sludge blanket depth is unknown.

Reliability: CF 100

- A. a) Obtain sludge blanket depth; b) Update data base; c) Run ASPX again.
- B & C. a) Obtain sludge blanket depth; b) Update data base.

Result Code: 61

Explanation: Waste Activated Sludge (WAS) rate is high.

Reliability: CF 98

- A. Check for stuck flowmeter on WAS pipeline.

Result Code: 62

Explanation: Waste Activated Sludge (WAS) rate is normal.

Reliability: CF 95

- A. Check for stuck flowmeter on WAS pipeline; c) Check for scum in clarifiers.

Result Code: 63

Explanation: F/M ratio is high.

Reliability: CF 90

- A. a) Check MLSS, TOC, and flowrate measurements; b) Inform management about high BOD loading; c) Eliminate high BOD source.
- B. Inform management about high BOD loading.

Result Code: 64

Explanation: F/M ratio is normal.

Reliability: CF 95

A. Check MLSS, TOC, and flowrate measurements.

Result Code: 65

Explanation: F/M ratio is unknown.

Reliability: CF 100

A & B. a) Obtain values for MLSS, TOC, and flowrate; b) Update data base; c) Run ASPX again.

C. a) Obtain values for MLSS, TOC, and flowrate; b) Update data base.

Result Code: 66

Explanation: Waste activated sludge suspended solids (Xw) is low.

Reliability: CF 80

A & B. a) Poor sludge settlement; b) check for possible sludge settlement problems; c) Investigate ways to capture solids escaping to the effluent.

Result Code: 67

Explanation: Waste activated sludge suspended solids (Xw) is normal.

Reliability: CF 95

A. Check solids analysis.

Result Code: 68

Explanation: Waste activated sludge suspended solids (Xw) is unknown.

Reliability: CF 100

A & B & C. a) Obtain value for Xw; b) Update data base; c) Run ASPX again.

Result Code: 69

Explanation: Waste Activated Sludge (WAS) rate is low.

Reliability: CF 95

A. a) Stop wasting sludge; b) Check for stuck flowmeter on WAS pipeline; c) Check for scum in clarifiers.

B & C. a) Reduce wasting to X gpm.

Result Code: 70

Explanation: Waste Activated Sludge (WAS) rate is unknown.

Reliability: CF 100

A & B & C. a) Obtain WAS rate; b) Update data base; c) Run ASPX again.

Result Code: 71

Explanation: Waste Activated Sludge (WAS) rate is zero.

Reliability: CF 100

Result Code: 72

Explanation: Sludge age is low.

Reliability: CF 98

A. Manually confirm sludge age calculations.

Result Code: 73

Explanation: Cannot calculate sludge age.

Reliability: CF 100

A & B & C. a) Obtain values for MLSS, Q_w , X_w , Q , and X_e ; b) Update data base; c) Run ASPX again.

Result Code: 74

Explanation: Increase in sludge age is requested to balance other abnormalities.

Reliability: CF 100

Result Code: 75

Explanation: There is no need to increase sludge age.

Reliability: CF 100

Result Code: 76

Explanation: Sludge age is normal.

Reliability: CF 95

A & B. Check MLSS, and flowrate measurements.

C. a) Sludge age might be too long; b) Observe pin floc in clarifier effluent.

Result Code: 77

Explanation: Blower intake filter is new.

Reliability: CF 90

- A. Check for malfunctioning blower gauges.

Result Code: 78

Explanation: Blower intake filter is old.

Reliability: CF 95

- A & B. Replace blower intake filters.

Result Code: 79

Explanation: Condition of blower intake filter is unknown.

Reliability: CF 100

- A & B. a) Check records to find out the last time the blower intake filter was replaced; b) Run ASPX again.

Result Code: 80

Explanation: Don't ask the condition of blower intake filters.

Reliability: CF 100

- A. There might be a problem with blower intake filters, run ASPX again with the "In-depth" option.
- B. There might be a problem with blower intake filters, run ASPX again with the "Regular" option.

Result Code: 81

Explanation: Blower suction pressure is normal.

Reliability: CF 95

- A & B. a) Undetermined blower problem; b) Decrease organic loading to compensate for low DO.

Result Code: 82

Explanation: Blower suction pressure is high.

Reliability: CF 95

- A. Check for malfunctioning suction pressure gauge.

Result Code: 83

Explanation: Blower suction pressure is unknown.

Reliability: CF 100

- A & B. a) Obtain value for the blower suction pressure; b) Update data base;
c) Run ASPX again.

Result Code: 84

Explanation: Don't ask the blower suction pressure.

Reliability: CF 100

- A. There might be a problem with blower suction pressure, run ASPX again with the "In-depth" option.
B. There might be a problem with blower suction pressure, run ASPX again with the "Regular" option.

Result Code: 85

Explanation: Part of discharge air is diverted.

Reliability: CF 100

- A & B & C. Close air blow-off valve so that all discharge air is directed to the aeration basins.

Result Code: 86

Explanation: It is unknown whether part of discharge air diverted or not.

Reliability: CF 100

- A & B. a) Find out if part of discharge air is diverted or not; b) Run ASPX again.

Result Code: 87

Explanation: Don't ask whether the discharge air is diverted or not.

Reliability: CF 100

- A. There might be a problem with air supply, run ASPX again with the "In-depth" option.
B. There might be a problem with air supply, run ASPX again with the "Regular" option.

Result Code: 88

Explanation: No part of the discharge air is diverted.

Reliability: CF 75

- A & B. High discharge drop, report to maintenance.

Result Code: 89

Explanation: Blower discharge pressure is low.

Reliability: CF 98

- A & B & C. Increase blower discharge pressure to X psi to obtain a proper DO distribution.

Result Code: 90

Explanation: Blower discharge pressure is normal.

Reliability: CF 90

- A. Check for malfunctioning blower discharge pressure gauge.

Result Code: 91

Explanation: Blower discharge pressure is unknown.

Reliability: CF 100

- A & B. a) Read blower discharge pressure gauge; b) Run ASPX again.

Result Code: 92

Explanation: Don't ask the blower discharge pressure.

Reliability: CF 100

- A. There might be a problem with discharge pressure drop, run ASPX again with the "In-depth" option.
- B. There might be a problem with discharge pressure drop, run ASPX again with the "Regular" option.

Result Code: 93

Explanation: Air control valve is partially or completely closed.

Reliability: CF 100

- A & B & C. Open air control valve completely to the open position.

Result Code: 94

Explanation: Position of air control valve is unknown.

Reliability: CF 100

- A & B. a) Check to find out the position of the air control valve; b) Run ASPX again.

Result Code: 95

Explanation: Don't ask the position of the air control valve.

Reliability: CF 100

- A. There might be a problem with the air control valve, run ASPX again with the "In-depth" option.
- B. There might be a problem with the air control valve, run ASPX again with the "Regular" option.

Result Code: 96

Explanation: Air control valve is completely open.

Reliability: CF 70

- A. a) Verify the position of the air control valve; b) inspect for stuck air control valve.

Result Code: 97

Explanation: Air flowrate is low.

Reliability: CF 98

- A & B & C. Increase air flowrate to X SCFM.

Result Code: 98

Explanation: Air flowrate is normal.

Reliability: CF 90

- A. Even though air flowrate seems normal, increase air flowrate to X (25%) SCFM.
- B. Even though air flowrate seems normal, increase air flowrate to X (10%) SCFM.
- C. Prepare to put additional blower to service.

Result Code: 99

Explanation: Air flowrate is unknown.

Reliability: CF 100

- A & B. a) Read air flowrate gauge; b) Run ASPX again.

Result Code: 100

Explanation: Don't ask air flowrate.

Reliability: CF 100

- A. There might be a problem with air flowrate, run ASPX again with the "In-depth" option.
- B. There might be a problem with air flowrate, run ASPX again with the "Regular" option.

Result Code: 101

Explanation: Diffusers are clean.

Reliability: CF 60

- A. a) Report to maintenance about possible diffuser breakage; b) If breakage is confirmed, turn down the blowers.
- B. Possible diffuser breakage.

Result Code: 102

Explanation: Diffusers are dirty.

Reliability: CF 70

- A & B. a) Take basin out of service; b) Clean and repair diffusers.
- C. Schedule maintenance for diffuser cleaning.

Result Code: 103

Explanation: Last diffuser cleaning is unknown.

Reliability: CF 100

- A & B. a) Consult maintenance records to find out the time the last diffuser cleaning was performed; b) Run ASPX again.

Result Code: 104

Explanation: Don't ask about diffuser cleaning.

Reliability: CF 100

- A. There might be a problem with air distribution, run ASPX again with the "In-depth" option.
- B. There might be a problem with air distribution, run ASPX again with the "Regular" option.

Result Code: 105

Explanation: Aeration surface is uniform.

Reliability: CF 50

- A. Confirm surface of aeration basin is uniform.

Result Code: 106

Explanation: Don't ask the appearance of the aeration surface.

Reliability: CF 100

- A. There might be a problem with mixing, run ASPX again with the "In-depth" option.

- B. There might be a problem with mixing, run ASPX again with the "Regular" option.

Result Code: 107

Explanation: Aeration surface is not uniform.

Reliability: CF 100

Result Code: 108

Explanation: No, we cannot divert part of the influent flow.

Reliability: CF 100

- A & B. Search for alternatives to reduce the incoming organic loading.

Result Code: 109

Explanation: Yes, we can divert part of the influent flow.

Reliability: CF 100

- A & B & C. Divert X gpm influent flow into temporary holding tanks for X hrs to compensate for high organic loading.

Result Code: 110

Explanation: Not sure if we can divert part of the influent flow.

Reliability: CF 100

- A & B. Consult supervisor to find out if you can divert part of influent flowrate to compensate for high organic loading.

Result Code: 111

Explanation: Don't ask whether we can divert part of the influent flow.

Reliability: CF 100

- A. There might be a problem with high organic loading, run ASPX again with the "In-depth" option.
- B. There might be a problem with high organic loading, run ASPX again with the "Regular" option.

Result Code: 112

Explanation: MLSS is low.

Reliability: CF 95

- A. Check MLSS analysis.

Result Code: 113

Explanation: MLSS seems normal.

Reliability: CF 90

- A. a) Report to management about high organic loading. b) Eliminate source of high organic loading.

Result Code: 114

Explanation: MLSS is unknown.

Reliability: CF 100

- A & B. a) Perform MLSS analysis; b) Update data base; c) Run ASPX again.
- C. a) Perform MLSS analysis; b) Update data base.

Result Code: 115

Explanation: F/M is low.

Reliability: CF 90

- A. Check MLSS, TOC, and flowrate measurements.
- B. Check TOC analysis.

Result Code: 116

Explanation: F/M ratio is high.

Reliability: CF 90

- A. a) Check MLSS, TOC, and flowrate measurements; b) Inform management about high BOD loading; c) Eliminate high BOD source.
- B. a) Check TOC analysis; b) Inform management about high BOD loading.
- C. Check TOC analysis.

Result Code: 117

Explanation: F/M ratio is unknown.

Reliability: CF 100

- A & B. a) Obtain values for MLSS, TOC, and flowrate; b) Update data base; c) Run ASPX again.
- C. a) Obtain values for MLSS, TOC, and flowrate; b) Update data base.

Result Code: 118

Explanation: F/M ratio is normal.

Reliability: CF 95

- A. Check MLSS and TOC analysis.

Result Code: 119

Explanation: More blowers are available.

Reliability: CF 100

- A & B & C. Put additional blower in service.

Result Code: 120

Explanation: Not sure whether more blowers are available or not.

Reliability: CF 100

- A & B & C. a) Find out if there are more available blowers that can be put in line to raise the dissolved oxygen concentration; b) Run ASPX again.

Result Code: 121

Explanation: Don't ask whether there are available blowers or not.

Reliability: CF 100

- A. There might be a problem with DO concentration, run ASPX again with the "In-depth" option.
- B. There might be a problem with DO concentration, run ASPX again with the "Regular" option.

Result Code: 122

Explanation: No additional blowers are available.

Reliability: CF 100

Result Code: 123

Explanation: Increase in DO level is requested.

Reliability: CF 100

- A & B. Although DO seems normal, increase DO to compensate for other abnormalities.

Result Code: 124

Explanation: Increase in DO level is not needed.

Reliability: CF 100

Result Code: 125

Explanation: DO is normal.

Reliability: CF 95

- A. a) Calibrate DO probes; b) Verify DO with portable DO meter.
- B. Verify DO with portable DO meter.

Result Code: 126

Explanation: DO concentration is unknown.

Reliability: CF 100

- A & B & C. a) Measure DO in aeration basins; b) Update data base; c) Run ASPX again.

Result Code: 127

Explanation: DO is low.

Reliability: CF 95

- A. a) Calibrate DO probes; b) Verify DO with portable DO meter.

Result Code: 128

Explanation: Influent pH is low.

Reliability: CF 95

- A. a) Confirm pH with a pH-paper; b) Inform management about possible acid spill within the refinery processes; c) find source and eliminate it.
- B. Inform management about possible acid spill within the refinery processes.

Result Code: 129

Explanation: Influent pH is normal.

Reliability: CF 90

- A. Confirm pH with a pH-paper.

Result Code: 130

Explanation: Influent pH is high.

Reliability: CF 95

- A. a) Confirm pH with a pH-paper; b) Inform management about possible caustic material spill within the refinery processes; c) find source and eliminate it.

- B. Inform management about possible caustic material spill within the refinery processes.

Result Code: 131

Explanation: Influent pH is unknown.

Reliability: CF 100

- A. a) Measure influent pH; b) Update data base; c) Run ASPX again.
- B & C. a) Measure influent pH; b) Update data base.

Result Code: 132

Explanation: Caustic addition rate is high.

Reliability: 95

- A & B & C. Reduce caustic addition rate to X gpm.

Result Code: 133

Explanation: Caustic addition rate is low.

Reliability: 98

- A & B & C. Increase caustic addition rate to X gpm.

Result Code: 134

Explanation: Caustic addition rate seems normal.

Reliability: 95

- A. a) Confirm caustic addition visually; b) Manually check the caustic level in the chemical tank; c) Check for stuck level indicator; d) Perform a manual caustic rate determination; e) If all fails, consider revising the pH controlling limits.
- B & C. a) Confirm caustic addition visually; b) Manually check the caustic level in the chemical tank; c) Check for stuck level indicator.

Result Code: 135

Explanation: Caustic addition is unknown.

Reliability: CF 100

- A & B. a) Obtain value for caustic addition rate; b) Update data base; c) Run ASPX again.

Result Code: 136

Explanation: Don't ask caustic addition rate.

Reliability: CF 100

- A. There might be a problem with caustic addition rate, run ASPX again with the "In-depth" option.
- B. There might be a problem with caustic addition rate, run ASPX again with the "Regular" option.

Result Code: 137

Explanation: Aeration basin pH is high.

Reliability: CF 95

- A. a) Calibrate pH electrodes; b) Check aeration basin pH with a pH-paper.

Result Code: 138

Explanation: Aeration basin pH is low.

Reliability: CF 95

- A. a) Calibrate pH electrodes; b) Check aeration basin pH with a pH-paper.

Result Code: 139

Explanation: Aeration basin pH is unknown.

Reliability: CF 100

- A & B & C. a) Measure pH in aeration basins; b) Update data base; c) Run ASPX again.

Result Code: 140

Explanation: Aeration basin pH is normal.

Reliability: CF 90

- A. a) Calibrate pH electrodes; b) Check aeration basin pH with a pH-paper.

Result Code: 141

Explanation: Phosphate addition is lower than calculated phosphate requirement.

Reliability: CF 98

- A & B & C. Increase phosphoric acid addition to X gpm to satisfy the nutrient requirements.

Result Code: 142

Explanation: Phosphate addition is higher than calculated phosphate requirement.

Reliability: CF 90

- A. a) Check phosphate analysis; b) Increase phosphoric acid addition rate to X gpm.
- B. Increase phosphoric acid addition rate to X gpm.
- C. Monitor the effluent phosphate concentration.

Result Code: 143

Explanation: Cannot calculate phosphate requirement.

Reliability: CF 100

- A & B. a) Obtain values for TOC; b) Update data base; c) Run ASPX again.
- C. a) Obtain values for TOC; b) Update data base.

Result Code: 144

Explanation: Phosphate addition is lower than calculated phosphate requirement.

Reliability: CF 98

- A & B & C. Increase phosphoric acid addition to X gpm to satisfy the nutrient requirements.

Result Code: 145

Explanation: Phosphate addition is higher than calculated phosphate requirement.

Reliability: CF 90

- A. a) Check phosphate analysis; b) Increase phosphoric acid addition rate to X (25%) gpm.
- B. Increase phosphoric acid addition rate to X (10%) gpm.
- C. Monitor the effluent phosphate concentration.

Result Code: 146

Explanation: Cannot calculate phosphate requirement.

Reliability: CF 100

- A & B. a) Obtain values for TOC; b) Update data base; c) Run ASPX again.
- C. a) Obtain values for TOC; b) Update data base.

Result Code: 147

Explanation: Phosphate addition rate is low.

Reliability: CF 90

- A. a) Inspect phosphoric acid addition visually; b) Check for empty phosphoric acid tank; c) Check for stuck phosphoric acid tank level indicator.

Result Code: 148

Explanation: Don't ask phosphoric acid addition rate.

Reliability: CF 100

- A. There might be a problem with phosphate concentration, run ASPX again with the "In-depth" option.
- B. There might be a problem with phosphate concentration, run ASPX again with the "Regular" option.

Result Code: 149

Explanation: Phosphate addition rate is unknown.

Reliability: CF 100

- A & B. a) Obtain value for phosphoric acid addition rate; b) Update data base; c) Run ASPX again.
- C. a) Obtain value for phosphoric acid addition rate; b) Update data base.

Result Code: 150

Explanation: Phosphate addition rate is normal.

Reliability: CF 90

- A. a) Inspect phosphoric acid addition visually; b) Check for empty phosphoric acid tank; c) Check for stuck phosphoric acid tank level indicator.
- B. Inspect phosphoric acid addition visually.

Result Code: 151

Explanation: Effluent phosphate concentration is low.

Reliability: CF 95

- A. Check phosphate analysis.

Result Code: 152

Explanation: Effluent phosphate concentration is unknown.

Reliability: CF 100

- A & B. a) Measure the effluent phosphate concentration; b) Update data base; c) Run ASPX again.
- C. a) Measure the effluent phosphate concentration; b) Update data base.

Result Code: 153

Explanation: Effluent phosphate concentration is normal.

Reliability: CF 95

- A. Check phosphate analysis.

Result Code: 154

Explanation: Yes, we can divert part of the influent flow.

Reliability: CF 100

- A & B & C. Divert X gpm influent flow into temporary holding tanks for X hrs to compensate for temperature abnormalities.

Result Code: 155

Explanation: No, we cannot divert part of the influent flow.

Reliability: CF 100

Result Code: 156

Explanation: Not sure if we can divert part of the influent flow.

Reliability: CF 100

- A & B. Consult supervisor to find out if you can divert part of the influent flowrate to compensate for the temperature abnormalities.

Result Code: 157

Explanation: Don't ask whether we can divert part of the influent flow.

Reliability: CF 100

- A. There might be a problem with the temperature, run ASPX again with the "In-depth" option.
- B. There might be a problem with the temperature, run ASPX again with the "Regular" option.

Result Code: 158

Explanation: Influent temperature is unknown.

Reliability: CF 100

- A. a) Measure influent temperature; b) Update data base; c) Run ASPX.

Result Code: 159

Explanation: Influent temperature is normal.

Reliability: CF 60

- A. Check temperature measuring equipment.

Result Code: 160

Explanation: Influent temperature is low.

Reliability: CF 80

- A. a) Find and eliminate the source of cold-water merge; b) Check for rainwater and blowdown water in the process flow; c) Check temperature measuring equipment.
- B. Find and eliminate the source of cold-water merge.

Result Code: 161

Explanation: Influent temperature is warmer than normal.

Reliability: CF 90

- A. a) Check temperature measuring equipment; b) Inform management about high temperature in the influent.
- B. Inform management about high temperature in the influent.

Result Code: 162

Explanation: Aeration basin temperature is too warm.

Reliability: CF 90

- A. a) Check temperature measuring equipment; b) Investigate ways to reduce temperature in the aeration basins to X°F.
- B. Check temperature measuring equipment.

Result Code: 163

Explanation: Aeration basin temperature is low.

Reliability: CF 60

- A. Check temperature measuring equipment.

Result Code: 164

Explanation: Aeration basin temperature is unknown.

Reliability: CF 100

- A & B. a) Measure influent temperature; b) Update data base; c) Run ASPX again.
- C. a) Measure influent temperature; b) Update data base.

Result Code: 165

Explanation: Aeration basin temperature is normal.

Reliability: CF 60

A. Check temperature measuring equipment.

Result Code: 166

Explanation: Pinpoint floc may be causing problems.

Reliability: CF 60

A & B. Check for pinpoint floc in the clarifiers.

Result Code: 167

Explanation: Ashing may be causing problems.

Reliability: CF 60

A & B. Check for very small size floats in the clarifiers.

Result Code: 168

Explanation: Straggler floc may be causing problems.

Reliability: CF 60

A & B. Check for small, fluffy flocs in the clarifiers.

Result Code: 169

Explanation: Floating sludge may be causing problems.

Reliability: CF 60

A & B. Check for rising sludge flocs in the clarifiers.

Result Code: 170

Explanation: White foaming at the clarifier surface is observed.

Reliability: CF 70

A & B. Check for "young sludge" (low MCRT).

Result Code: 171

Explanation: Thick brown foaming at the clarifier surface is observed.

Reliability: CF 70

A & B. Check for irregular loading rates (low F/M).

Result Code: 172

Explanation: Black foaming at the clarifier surface is observed.

Reliability: CF 70

A & B. Check for anaerobic conditions in the clarifiers.

Result Code: 173

Explanation: Light brown foaming at the clarifier surface is observed.

Reliability: CF 70

A & B. Cause of sludge settling problem is unknown.

Appendix C

Questionnaire Results

Number of Responds = 13

Position on staff:

Support Supervisor = 2

RPT = 8

ETP Operator = 2

Division Environmental Coordinator-Operations = 1

Practical experience operating Activated Sludge Plants:

< 1 yr = 1

1-2 yrs = 2

3 yrs = 4

4 yrs = 1

7 yrs = 1

12+ yrs = 4

Questions & Answers:

1. When the plant experiences a *nitrification failure*, how often do you know what is causing the problem?

Valid Responds: 13 out of 13

0% (0)* Have very little idea.

70% (9) Have some idea but not enough to correct the problem right away.

15% (2) Have a pretty good idea about it. I can find the primary cause and fix the problem in less than an hour.

0% (0) I usually know what is causing the problem and can solve it within minutes.

15% (2) Other: Can diagnose early but it takes time to correct it.

*The number preceding the "%" indicates the valid responds percentage for that option. The number in parenthesis shows the actual number of people who chose that specific option.

2. What percentage of time do you use the following methods to *diagnose* or *detect* nitrification problems?

Observing the control panel Valid Responds: 8 out of 13

50% (4) < 20 percent of the time
50% (4) =20..40 percent of the time

Visual observation on field Valid Responds: 7 out of 13

29% (2) < 20 percent of the time
57% (4) =20..40 percent of the time
14% (1) =40..60 percent of the time

Instinct coming from being experienced Valid Responds: 8 out of 13

38% (3) < 20 percent of the time
38% (3) =20..40 percent of the time
25% (2) =40..60% percent of the time

Analytical/Lab measurement results Valid Responds: 13 out of 13

31% (4) =20..40 percent of the time
15% (2) =40..60 percent of the time
23% (3) =60..80 percent of the time
15% (2) =80..90 percent of the time
15% (2) =90..100 percent of the time

Other: Some special conditions ?

3. Generally, how do you *diagnose/detect* the following problems? Please, check all that apply.

	Field Measurement	Control Panel	Visual Observation	Experience
Low DO level	37% (11)	43% (13)	7% (2)	13% (4)
Nitrification lost	82% (13)	6% (1)	6% (1)	6% (1)
Low MLSS level	57% (13)	0% (0)	30% (7)	13% (3)
Clogged diffusers	13% (2)	20% (3)	53% (8)	13% (2)
High grease/oil in basins	44% (12)	7% (2)	33% (9)	15% (4)
High organic loading	58% (11)	26% (5)	11% (2)	5% (1)
High sulfide in "Feed"	72% (13)	11% (2)	0% (0)	17% (3)
Hydraulic overloading	29% (6)	19% (4)	29% (6)	24% (5)
Improper pH	54% (13)	46% (11)	0% (0)	0% (0)
Leaks in air piping	15% (3)	20% (4)	50% (10)	15% (3)
Nocardia foaming	0% (0)	0% (0)	100% (3)	0% (0)
Other foaming	6% (1)	6% (1)	77% (13)	12% (2)
Improper PO ₄ addition	81% (13)	6% (1)	6% (1)	6% (1)
Septic conditions	17% (3)	6% (1)	44% (8)	33% (6)
Too high DO level	37% (11)	37% (11)	17% (5)	10% (3)
Too high MLSS level	59% (13)	9% (2)	23% (5)	9% (2)
Toxic shock	36% (8)	23% (5)	18% (4)	23% (5)
Rising bubbles in clarifiers	6% (1)	0% (0)	75% (12)	19% (3)
Faulty valve	17% (4)	13% (3)	38% (9)	33% (8)
Faulty measuring equipment	27% (10)	22% (8)	22% (8)	30% (11)
Problems with liquid pipelines	16% (5)	19% (6)	36% (11)	29% (9)
Broken pump	14% (5)	25% (9)	33% (12)	28% (10)
Problems with air blower	20% (9)	29% (13)	29% (13)	22% (10)

4. Are you able to distribute the air flow to the different sections of the aeration basins for example by introducing more air at the "head end" of the basins rather than the "exit end" of the basins or vice versa?

Valid Responds: 13 out of 13

62% (8) Yes
31% (4) No
8% (1) Not Sure

5. Would you be able to identify *Nocardia* foaming without microscopic examination?

Valid Responds: 12 out of 13

8% (1) Yes
33% (4) No
58% (7) Not Sure

If yes, how:

Valid Responds: 2 out of 13

50% (1) Color & concentration of the foam
50% (1) Dark Scummy Foam

6. Besides the following methods, how would you check for clogged diffusers?
- Increase in air blower head pressure
 - Abnormal mixing pattern
 - Non-uniform DO concentration

Valid Responds: 9 out of 13

67% (6) Take basin out of service
22% (2) Shut down air valve & measure DO
11% (1) Monitor COD & SO₃ out

7. Are you able to measure DO at different locations in the aeration basins such as at the four corners, 5 ft. below surface *etc.*

Valid Responds: 13 out of 13

85% (11) Yes
8% (1) No
8% (1) Not Sure

8. In case of a pump failure with no backup unit, how fast can your maintenance personnel put a new pump on-line?

Valid Responds: 12 out of 13

- 8% (1) 1-3 hours
- 25% (3) 4-5 hours
- 8% (1) 6 hours
- 8% (1) 8 hours
- 8% (1) 12 hours
- 8% (1) 8-24 hours
- 17% (2) 1 day - 1 month
- 8% (1) 6 months
- 8% (1) Too long

9. Are you equipped to perform a "leak detection test" in air piping?

Valid Responds: 13 out of 13

- 54% (7) Yes
- 31% (4) No
- 16% (2) Not Sure

If yes, explain the procedure:

Valid Responds: 8 out of 13

- 50% (4) Use soap bubbles
- 13% (1) Use special leak detector solution
- 25% (2) Close discharge valves & observe pressure
- 13% (1) Visually inspect

10. Has there been any time that you needed a lab result but it wasn't available for you and you were forced to make a decision in the absence of this data?

Valid Responds: 13 out of 13

- 46% (6) Yes
- 54% (7) No

If yes, what is your suggestion to improve the data collection system?

Valid Responds: 5 out of 13

- 20% (1) Cut down on routine measurements & focus on more important ones
- 40% (2) Full-time Lab Tech. & more accurate results
- 20% (1) Faster results & Lab Tech.
- 20% (1) Phenol analysis needs improvement

11. With problems in the past, would it have been helpful to have detailed measurements, such as a typical value, graphical relationships to other parameters, frequency of upsets and so forth?

Valid Responds: 12 out of 13

58% (7) Yes

0% (0) No

42% (5) Not Sure

If yes, what specific measurements were you seeking?

14% (1) Number of upsets

14% (1) NH₄, Oil, Temperature reading

14% (1) NH₃ vs nitrification loss & MLSS vs removal efficiency

27% (2) Effects of parameters on the activated sludge process

14% (1) COD concentration tracking

14% (1) Current measurements are sufficient

12. If you were to be provided the following measurements about a parameter, how valuable would they be to you?

	Must Have					
	5	4	3	2	1	0
Typical value	54% (7)	31% (4)	15% (2)	0% (0)	0% (0)	0
Value considered "abnormal"	54% (7)	15% (2)	23% (3)	0% (0)	8% (1)	0
Graphical relationships to other parameters	31% (4)	15% (2)	36% (6)	0% (0)	8% (1)	0
Estimated future value	17% (2)	25% (3)	33% (4)	8% (1)	17% (2)	0
Last week's pattern	33% (4)	33% (4)	25% (3)	0% (0)	8% (1)	0
How often the current problem has occurred in the past	46% (6)	15% (2)	31% (4)	8% (1)	0% (0)	0
<u>Value desired now</u>	100% (1)	0	0	0	0	0

13. What following *spare* equipment do you have at the treatment plant?

Yes	No	Not Sure	
69% (9)	31% (4)	0% (0)	Portable pH-meter or other alternative pH-meters.
92% (12)	8% (1)	0% (0)	Spare full capacity air blower.
100% (13)	0% (0)	0% (0)	Portable DO-probe & DO-meter.
39% (5)	62% (8)	0% (0)	Emergency pH delivery system.
31% (4)	69% (9)	0% (0)	Alternative PO ₄ delivery system.
0% (0)	92% (12)	8% (1)	Lab setup to measure PO ₄ with at least two methods.
23% (3)	69% (9)	8% (1)	Alternative NH ₄ , NO ₂ and NO ₃ measurement equipment.

14. In case of emergency, do you know how to calibrate and operate the following instruments?

Yes	No	Little bit but not enough	
85% (11)	15% (2)	0% (0)	Portable pH-meter.
85% (11)	8% (1)	8% (1)	Portable DO-meter.
23% (3)	77% (10)	0% (0)	Portable Ammonia electrode.
0% (0)	100% (13)	0% (0)	Portable Nitrate electrode.

15. Please, indicate how often you experience the following problems in this treatment plant? (Leave blank to indicate "Never").

	Every-day	3 times a week	Once a week	Once a month	Once a year
Clogged Diffusers	0% (0)	0% (0)	0% (0)	0% (0)	54% (7)
Cold temperature (<77 F)	0% (0)	0% (0)	8% (1)	0% (0)	15% (2)
Bulking Sludge	0% (0)	8% (1)	8% (1)	31% (4)	23% (3)
Cloudy effluent	8% (1)	0% (0)	31% (4)	31% (4)	31% (4)
High grease/oil in basins	8% (1)	15% (2)	0% (0)	69% (9)	0% (0)
High organic loading	8% (1)	15% (2)	8% (1)	62% (8)	0% (0)
High sulfide in "Feed"	0% (0)	8% (1)	23% (3)	69% (9)	0% (0)
Hydraulic overloading	8% (1)	0% (0)	0% (0)	31% (4)	23% (3)
Improper pH	0% (0)	0% (0)	8% (1)	46% (6)	31% (4)
Leaks in air piping	0% (0)	0% (0)	0% (0)	0% (0)	39% (5)
Nocardia foaming	0% (0)	0% (0)	0% (0)	8% (1)	0% (0)
Other foaming	8% (1)	0% (0)	15% (2)	62% (8)	8% (1)
Improper PO ₄ addition	0% (0)	0% (0)	0% (0)	46% (6)	39% (5)
Septic conditions	0% (0)	0% (0)	0% (0)	15% (2)	23% (3)
Too high DO level	0% (0)	0% (0)	15% (2)	77% (10)	8% (1)
Too high MLSS level	0% (0)	0% (0)	8% (1)	85% (11)	8% (1)
Toxic shock	0% (0)	0% (0)	8% (1)	31% (4)	39% (5)
Rising bubbles in clarifiers	8% (1)	0% (0)	15% (2)	31% (4)	39% (5)
Faulty valve	0% (0)	0% (0)	0% (0)	23% (3)	62% (8)
Faulty measuring equipment	0% (0)	0% (0)	46% (6)	39% (5)	8% (1)
Problems with liquid pipelines	0% (0)	0% (0)	8% (1)	15% (2)	46% (6)
Broken pump	0% (0)	0% (0)	8% (1)	31% (4)	54% (7)
Problems with air blower	0% (0)	0% (0)	0% (0)	15% (2)	62% (8)
High Sludge Blanket Depth	0% (0)	0% (0)	0% (0)	15% (2)	0% (0)
Plugged upcomers ?	0% (0)	0% (0)	0% (0)	8% (1)	0% (0)
Too low MLSS	0% (0)	0% (0)	0% (0)	8% (1)	0% (0)
Sludge Blanket problems	0% (0)	0% (0)	0% (0)	8% (1)	0% (0)

16. Please, outline the procedural steps you would take in order to perform the following actions.

Decrease WAS: Valid Responds: 7 out of 13

- 29% (2) Waste
- 57% (3) No waste
- 14% (1) Manipulate valve

Decrease F/M ratio: Valid Responds: 13 out of 13

- 7% (1) Waste
- 7% (1) Check MLSS & other parameters
- 31% (4) Reduce flowrate/load
- 46% (6) Control COD & MLSS
- 7% (1) Increase recycle flowrate

Check caustic delivery system: Valid Responds: 13 out of 13

- 7% (1) Call supervisor
- 62% (8) Check visually
- 31% (4) Check pH

Check PO₄ delivery system: Valid Responds: 13 out of 13

- 7% (1) Call Supervisor
- 7% (1) Check pump pressure gauge
- 62% (8) Check residual PO₄
- 23% (3) Check visually

Check pH with portable pH-meter: Valid Responds: 12 out of 13

- 17% (2) Never done it
- 75% (9) Described proper procedure
- 8% (1) Take sample to Lab

Check for leaks in air piping: Valid Responds: 13 out of 13

- 46% (6) Use soap
- 7% (1) Observe DO concentration
- 31% (4) Check visually
- 7% (1) Observe blower pressure gauge

Increase sludge age: Valid Responds: 13 out of 13

- 69% (9) Decrease waste flowrate
- 23% (3) Don't know
- 7% (1) Decrease recycle flowrate

17. Current DO reading is 2.0 mg/L. If I tell you to raise the DO level to 3.8 mg/L, how would you adjust the blower? Explain in detail.

Valid Responds: 13 out of 13

- 92% (12) Adjust air valves accordingly
8% (1) Close exhaust valve

18. Are you able to answer the following questions using *only* visual observations made in the field?

Yes	Maybe	No	
69% (9)	31% (4)	0% (0)	Difference between "Stiff white foam," "Sudsy foam," "Shiny dark brown foam."
69% (9)	8% (1)	23% (3)	Difference between "Rotten eggs odor," "Sweet odor," "Earthy odor."
69% (9)	31% (4)	0% (0)	Difference between "Light tan color sludge," "Light brown color sludge," "Brown color sludge," "Black color sludge."
17% (2)	58% (7)	25% (3)	Difference between "Solids washout," "Bulking sludge," "Clumping/rising sludge," "Cloudy secondary effluent," "Ashing," "Pinpoint floc in clarifier," "Straggler floc."
54% (7)	8% (1)	39% (5)	Difference between "High MLSS," "Low MLSS."

19. Being "0%" as "very unlikely" and "100%" as "very possible," what would you say the likelihood of the following episodes happening is.

Episode I:

DO level is normal, sludge age is 6 days, no nitrification is occurring and MLSS analysis shows 5,000 mg/L.

(IMPORTANT!! The sum of all the percentages need not to be equal to 100%)
(Think of each "Case" as an individual event!)

Likelihood of:

Case 1: The TSS analysis being wrong **Valid Responds: 12 out of 13**

- 58% (7) < 20 percent of the time
8% (1) =20..40 percent of the time
17% (2) =40..60 percent of the time
17% (2) =60..80 percent of the time

Case 2: The DO probe being wrong Valid Responds: 12 out of 13

33% (4) < 20 percent of the time
42% (5) =20..40 percent of the time
8% (1) =40..60 percent of the time
17% (2) =60..80 percent of the time

Case 3: Possible toxic shock Valid Responds: 13 out of 13

8% (1) < 20 percent of the time
39% (5) =20..40 percent of the time
8% (1) =40..60 percent of the time
15% (2) =60..80 percent of the time
23% (3) =80..90 percent of the time
8% (1) =90..100 percent of the time

Case 4: Other Valid Responds: 4 out of 13

25% (1) =20..40 percent of the time, Not Sure
25% (1) =40..60 percent of the time, Number recording is bad
50% (2) =60..80 percent of the time, COD value & Sludge age is low

Episode II:

DO is low, about 2 mg/L, we have all blowers working on maximum capacity, MLSS is normal and organic loading is also normal.

Likelihood of:

Case 1: Clogged diffusers Valid Responds: 11 out of 13

27% (3) < 20 percent of the time
0% (0) =20..40 percent of the time
55% (6) =40..60 percent of the time
0% (0) =60..80 percent of the time
18% (2) =80..90 percent of the time

Case 2: The DO probe being wrong Valid Responds: 13 out of 13

8% (1) < 20 percent of the time
23% (3) =20..40 percent of the time
15% (2) =40..60 percent of the time
31% (4) =60..80 percent of the time
15% (2) =80..90 percent of the time
8% (1) =90..100 percent of the time

Case 3: Leaks in air piping Valid Responds: 8 out of 13

75% (6) < 20 percent of the time
13% (1) =20..40 percent of the time
13% (1) =40..60 percent of the time

Case 4: Air blower being broken Valid Responds: 9 out of 13

78% (7) < 20 percent of the time
0% (0) =20..40 percent of the time
11% (1) =40..60 percent of the time
11% (1) =60..80 percent of the time

Case 5: Other Valid Responds: 4 out of 13

25% (1) =20..40 percent of the time, COD is high
25% (1) =40..60 percent of the time, Air is directed to other sources
50% (2) =60..80 percent of the time, High conc. of toxics & Oxygen scavengers

20. a) Which computer do you like better?

Valid Responds: 12 out of 13

92% (11) Macintosh
0% (0) IBM
8% (1) None

b) What quality(s) do you like the best about your computer preference?

Valid Responds: 11 out of 13

64% (7) Only one to use here
36% (4) User-friendly

21. If I were able to provide you with a computer tool that would perform the following characteristics, which one would you choose?
Computer program which will help me:

Valid Responds: 13 out of 13

15% (2) only diagnose/detect the problem.

8% (1) just tell me what is causing the problem.

15% (2) detect the problem and identify positive probable causes as well.

62% (8) give complete advice on solving the problem, including problem detection, problem identification, solutions to the problem and preventive actions that I can take to eliminate future problems.

22. If a computer tool such as the one you chose above could be provided to you, how often would you use it?

Valid Responds: 13 out of 13

- 31% (4) Everyday.
0% (0) Every other day.
15% (2) Occasionally.
46% (6) Only when there is a problem with the plant operations.
0% (0) Only in case of a *severe* problem at the plant.
0% (0) Never.
8% (1) Other Use it as trainer & whenever a problem with the plant

23. Again, if such a program as above is provided to you, how likely are you going to trust the advice given by the computer?

Valid Responds: 16 out of 13

- 0% (0) I would unquestionably take the advice and do whatever the computer says.
- 0% (0) I will take the computer's advice but I probably will take its' advice with a little bit of a doubt.
- 50% (8) I would confirm the advice with other members of the staff before taking any action.
- 44% (7) I would put it on trial, and see if it really is reliable.
- 0% (0) Even though I would use the computer to obtain additional ideas, I probably will not consider its' advice seriously and would seldom use its advice.
- 0% (0) I would never trust a computer.
- 6% (1) Other Compare it with my experience

24. Which one of the following alternatives would you prefer to have while using a computer?

Valid Responds: 13 out of 13

0% (0) I like everything explained in "text." No pictures or graphics are necessary.

92% (12) I like a combination of pictures and text. The text should thoroughly explain the pictures and may give additional information.

8% (1) I don't like reading pages of written comments. Give me only pictures and graphics with very little text.

25. How much time are you willingly to spend on the computer consultation sessions?

Valid Responds: 9 out of 13

33% (3) 10-15 min. every time I sit in front of the computer.

33% (3) 20 min. every time I sit in front of the computer.

34% (3) Whatever it takes.

Cont'd next page.

Valid Responds: 10 out of 13

20% (2) 1/2 hr. per day.

40% (4) 1 hr. per day.

10% (1) 2 hrs. per day.

30% (3) Whatever it takes.

26. Would you find it useful to have a *plant simulator* on which you could examine the outcome and progresses of various hypothetical and real-time plant upsets.

Valid Responds: 13 out of 13

69% (9) Yes

15% (2) No

15% (2) Not Sure

If Yes, answer the following question. Otherwise, skip the next question.

27. If we were to provide you with a *plant simulator* that will simulate your treatment plant, how willing are you to do the following? Please, check all that apply.

Valid Responds: 18 out of 13

- 11% (2) Every time I change or adjust an instrument in the real-time plant, I would first test it with the computer to see the simulation outcome.
- 33% (6) I would only use the computer simulation when I have a doubt about my decision.
- 44% (8) During my spare time, I would play with the computer simulation by giving hypothetical values and observe what happens in the simulation.
- 11% (2) I would not use the computer simulation because it does not give solutions to my current problems.
- 0% (0) I would not use it because I would spend more time figuring out how to use the computer than seeking the answer elsewhere.

28. What kind of information, diagnosis, or other service would you like to receive from a computer? Please, include all remarks.

Valid Responds: 10 out of 13

- 30% (3) Included Information is sufficient
- 10% (1) Variable interrelations
- 10% (1) Variable tracking
- 10% (1) Toxic chemical list and effects
- 10% (1) Easy to use
- 10% (1) Glossary of terms
- 10% (1) Low-high values for variables
- 10% (1) Computer is not going to be used

Appendix D

Determination of Nitrification State Value

Program State_Determination;

Uses Menu4, Crt, Graph, GrDrivers, GrFonts, Axis, Curvefit;

```
Var gd, gm, Low, Hi, MAX: Integer;
  i, j, n, Order, Ftype: Integer;
  Xmin, Xmax, Ymin, Ymax: Real;
  Ey, Ey2, R, D, Std1, Std2: Real;
  Increment, Func: Real;
  Fname, Msg: AnyString;
  X, Y, EsasX, EsasY, Fx, Fy: RealArray;
  XX, YY: RealArray;
  AraX, AraY: Integer;
  Renk: Byte;
  st, dt: Text;
  Diff, y5: Real;
  RealState: String[50];
  STD: Integer;
  top: Real;
  SPL, PL, PPL, OPT: Array [1..4] of Real; { SPL= Severe Problem Limits }
  P: Array[1..3] of Integer;
  State: Array [1..4] of Integer;
  Slope, LastDay, E5, MaxVar: Integer;
  R_Slope, R_LastDay, R_E5: Integer;
  WF, CurSlope, CurLastDay, CurE5: Real;
  VarH1, VarH2, VarL1, VarL2: Integer;
  Violation: Integer;
  Limit: Array [1..3] of Real;
  FinalValue: Real;
```

```
Const Font: Word= DefaultFont;
      Size: Word= 1;
```

```
Procedure Init;
begin; gd:= VGA; gm:= VGAHi;
      InitGraph(gd, gm, '');
      SetColor(White);
      AraX:= 5;
      AraY:= 10;
      Renk:= Blue;
end;
```

```

Procedure Input;
  begin; Assign(dt,'5DAY-NH3.DAT'); Reset(dt);
    i:=0;
    While NOT EOF(dt) do begin;
      i:= Succ(i);
      Readln(dt,X[i],Y[i]);
    end;
  Close(dt);
  n:= i;
  For i:= 1 to n do begin; EsasX[i]:= X[i];
    EsasY[i]:= Y[i]; end;
  ShellsortR(X,n);
  Xmax:= X[n]; Xmin:= X[1];
  ShellsortR(Y,n);
  Ymax:= Y[n]; Ymin:= Y[1];

  For i:= 1 to n do begin; X[i]:= EsasX[i];
    Y[i]:= EsasY[i]; end;

  Ftype:= 1;
  Order:= 1;
end;

```

```

Procedure Variance;
  begin; VarH1:= Trunc(Result[2]*1+Result[1]);
    VarH2:= Trunc(Result[2]*5+Result[1]);
    If (VarH1+MaxVar) > MAX Then VarH1:= MAX Else VarH1:= VarH1+MaxVar;
    If (VarH2+MaxVar) > MAX Then VarH2:= MAX Else VarH2:= VarH2+MaxVar;
    VarL1:= Trunc(Result[2]*1+Result[1]);
    VarL2:= Trunc(Result[2]*5+Result[1]);
    If (VarL1-MaxVar) < 0 Then VarL1:= 0 Else VarL1:= VarL1-MaxVar;
    If (VarL2-MaxVar) < 0 Then VarL2:= 0 Else VarL2:= VarL2-MaxVar;
end;

```

```

Procedure CheckState;
  begin; Assign(st,'State.LMT'); Reset(st);
    Readln(st,MAX);
    Readln(st,STD);
    For i:= 1 to 3 do Readln(st,P[i]);
    For i:= 1 to 4 do Readln(st,State[i]);
    Readln(st,Slope);
    Readln(st,LastDay);
    Readln(st,E5);
    Readln(st,MaxVar);
    Close(st);
    CurSlope:= ((Result[2]*5+Result[1]) - (Result[2]*1+Result[1]));
    CurLastDay:= Result[2]*5+Result[1];
    top:= 0;
    For i:= 1 to 5 do top:= top+(Result[2]*i+Result[1]);
    CurE5:= top;
    R_Slope:= State[4];
end;

```

```

For i:= 1 to 3 do
  If CurSlope >= (P[i]-6) Then begin; R_Slope:= State[i];
    i:= 3; end;
R_LastDay:= State[4];
For i:= 1 to 3 do
  If CurLastDay >= P[i] Then begin; R_LastDay:= State[i];
    i:= 3; end;
R_E5:= State[4];
For i:= 1 to 3 do begin; top:= 0;
  For j:= 1 to 5 do top:= top+((P[i]-6)*(j-1)/4)+6;
  If CurE5 >= top Then begin; R_E5:= State[i];
    i:= 3; end; end;
For i:= 1 to 3 do
  Limit[i]:= (State[i+1]*(Slope+LastDay)+State[i]*E5)/State[1];

FinalValue:= (R_Slope*Slope+R_LastDay*LastDay+R_E5*E5)/State[1];

If FinalValue > Limit[1] Then RealState:= 'Severe Problem';
If (FinalValue <= Limit[1]) and (FinalValue > Limit[2]) Then RealState:= 'Problem';
If (FinalValue <= Limit[2]) and (FinalValue > Limit[3]) Then RealState:= 'Potential Problem';
If (FinalValue <= Limit[3]) Then RealState:= 'Optimum Operations';

Variance;
Violation:= 0;
For i:= 1 to n do
  If (EsasY[i] > (Result[2]*EsasX[i]+Result[1]+MaxVar)) or
    (EsasY[i] < (Result[2]*EsasX[i]+Result[1]-MaxVar)) Then
    Violation:= Violation+1;

If ((RealState= 'Optimum Operation') or
  (RealState= 'Potential Problem')) and
  (Violation >= 4) Then RealState:= 'Problem';
end;

Procedure Statistics;
begin; Ey2:= 0; Ey:= 0;
  For i:= 1 to n do begin; Ey2:= Ey2+Y[i]*Y[i];
    Ey := Ey +Y[i]; end;
  Ey:= Ey/n; Std1:= Sqrt(Ey2/n-Ey*Ey);

  For i:= 1 to n do begin;
    If Ftype in [2,4] Then Fx[i]:= ln(X[i])
      Else Fx[i]:= X[i];
    Func:= 0;
    For j:= 1 to (Order+1) do Func:= Func+Result[j]*Power(Fx[i],j-1);
    If Ftype in [3,4] Then Fy[i]:= Exp(Func)
      Else Fy[i]:= Func;
  end;
  Ey2:= 0; Ey:= 0;
  For i:= 1 to n do begin; Ey2:= Ey2+Fy[i]*Fy[i];
    Ey := Ey +Fy[i]; end;

```



```

    Ey: = Ey/n; Std2: = Sqrt(Ey2/n-Ey*Ey);
    If Std2 > Std1 Then R: = Std1/Std2
        Else R: = Std2/Std1; D: = R*R*100;
end;

Procedure GenerateFunction;
begin; Increment: = (Xmax-Xmin)/50;
    Fx[1]: = Xmin;
    Func: = 0;
    For j: = 1 to (Order+1) do
        If Ftype in [2,4] Then Func: = Func + Result[j]*Power(ln(Fx[1]),j-1)
            Else Func: = Func + Result[j]*Power( Fx[1] ,j-1);
    If Ftype in [3,4] Then Fy[i]: = Exp(Func)
        Else Fy[i]: = Func;

    For i: = 2 to 51 do begin;
        Fx[i]: = Fx[i-1]+Increment;
        Func: = 0;
        For j: = 1 to (Order+1) do
            If Ftype in [2,4] Then Func: = Func + Result[j]*Power(ln(Fx[i]),j-1)
                Else Func: = Func + Result[j]*Power( Fx[i] ,j-1);
            If Ftype in [3,4] Then Fy[i]: = Exp(Func)
                Else Fy[i]: = Func;

            If Fy[i] > Ymax Then Ymax: = Fy[i];
            If Fy[i] < Ymin Then Ymin: = Fy[i];
        end;
    end;

Procedure DrawLine(x1,y1,x2,y2: Integer; Color:Byte);
begin; XX[1]: = x1; YY[1]: = y1;
    XX[2]: = x2; YY[2]: = y2;
    DrawXY(XX,YY,2,Color);
end;

BEGIN
    Input;
    Curve(X,Y,n,Ftype,Order);
    CheckState;
    Statistics;
    GenerateFunction;

    Assign(dt,'STATE.VAL'); Rewrite(dt);
    WriteLn(dt,''' + RealState + ''');
    Close(dt);

    Init;
    SetColor(Red);
    SetBkColor(Blue);
    SetTextStyle(1,Horizdir,3);

```

```

Msg:= 'Effluent Nitrification Data';
OutTextxy(Trunc(GetMaxX/2-TextWidth(Msg)/2),TextHeight(Msg),Msg);
SetColor(LightCyan);
SetTextStyle(1,Horizdir,2);
Msg:= 'Nitrification State: ' + RealState;
OutTextxy(Trunc(GetMaxX/2-TextWidth(Msg)/2),(3*TextHeight(Msg)),Msg);
SetColor(Yellow);
SetTextStyle(0,0,0);
Str(R:5:3,Msg);
Msg:= 'Correlation Coefficient = ' +Msg;
OutTextxy(Trunc(GetMaxX/2-TextWidth(Msg)/2),(10*TextHeight(Msg)),Msg);
Str(((n-Violation)/n*100):3:0,Msg);
Msg:= Msg + '% of data within the Variance Range';
OutTextxy(Trunc(GetMaxX/2-TextWidth(Msg)/2),(11*TextHeight(Msg)),Msg);

SetColor(GetMaxColor);
Xaxis(Xmin,Xmax,AraX,'Today (Day 5) & Last 4 days',Font,Size);
Yaxis(0,Max-10,10,'Effluent NH4+ Concentration (mg/L)',Font,Size);

DrawXYPoint(EsasX,EsasY,n,5,2,LightBlue);
DrawLine(1,6,5,P[1],DarkGray);
DrawLine(1,6,5,P[2],DarkGray);
DrawLine(1,6,5,P[3],DarkGray);

SetLineStyle(DashedLn,0,ThickWidth);
DrawLine(0,STD,6,STD,LightRed);

SetLineStyle(SolidLn,0,NormWidth);
DrawXy(Fx,Fy,51,LightCyan);
SetLineStyle(DottedLn,0,NormWidth);
DrawLine(1,VarH1,5,VarH2,LightCyan);
DrawLine(1,VarL1,5,VarL2,LightCyan);

Getkey;
ClearDevice;
CloseGraph;
RestoreCrtMode;
END.

```

Appendix E

ASPX Knowledge Base Rules

```
DOMAIN :: (:TAB :ATTR (RED) "A" :ATTR (GREEN)
"ctivated " :ATTR (RED) "S" :ATTR (GREEN) "hudge "
:ATTR (RED) "P" :ATTR (GREEN) "roceas e" :ATTR (RED)
"X" :ATTR (GREEN) "pert :: " :ATTR (RED) "ASPX" )
ROOT FRAME :: ASPX
```

```
=====
Global KB data
=====
```

```
FRAME STRUCTURE ::
ASPX
NITRIFICATION
INITIAL
AMMONIA
SUB-NH4
SA
SUB-SA
DISSOLVED-OXYGEN
SUB-DO
PH
SUB-PH
PO4
SUB-PO4
TEMPERATURE
SUB-TEMP
SETTLING-PROBLEMS
```

```
KB Files :: (NITRIFICATION "VER7-B.k15" INITIAL "VER7-
B.k14" PH "VER7-B.k13" SA "VER7-B.k12" DISSOLVED-
OXYGEN "VER7-B.k11" AMMONIA " VER7-B.k10" PO4
"VER7-B.k9" TEMPERATURE "VER7-B.k8" SUB-NH4 " VER7-
B.k7" SUB-SA "VER7-B.k6" SUB-DO "VER7-B.k5" SUB-PH "
VER7-B.k4" SUB-PO4 "VER7-B.k3" SUB-TEMP "VER7-B.k2"
SETTLING-PROBLEMS "VER7-B.k1" )
```

```
Parameter groups :: (SETTLING-PROBLEMS-PARMS SUB-
TEMP-PARMS SUB-PO4-PARMS SUB-PH-PARMS SUB-DO-
PARMS SUB-SA-PARMS SUB-NH4-PARMS TEMPERATURE-
PARMS PO4-PARMS AMMONIA-PARMS DISSOLVED-
OXYGEN-PARMS SA-PARMS PH-PARMS INITIAL-PARMS
NITRIFICATION-PARMS ASPX-PARMS ) Rule groups ::
(SETTLING-PROBLEMS-RULES SUB-TEMP-RULES SUB-PO4-
RULES SUB-PH-RULES SUB-DO-RULES SUB-SA-RULES SUB-
NH4-RULES TEMPERATURE-RULES PO4-RULES AMMONIA-
RULES DISSOLVED-OXYGEN-RULES SA-RULES PH-RULES
INITIAL-RULES NITRIFICATION-RULES ASPX-RULES
META-RULES ) Number of rules :: 338
```

```
Number of meta-rules :: 1
Variables :: (DOMAIN RUNTIME)
TEXTAGS :: ()
Functions :: ()
```

```
=====
Frame :: ASPX
=====
IDENTIFIER :: "ASPX-"
GOALS :: (STATE RC SSP YAZ)
DISPLAYRESULTS :: (RC SSP)
PARMGROUP :: ASPX-PARMS
RULEGROUPS :: (ASPX-RULES)
OFFSPRING :: (NITRIFICATION SETTLING-PROBLEMS)
ASPX-PARMS :: (BEP-L BOD5/TKN BREAK C/N-H CALC-PH
CAUSTIC-H CAUSTIC-L CHECK-SSP CODIN CODOUT
CURDO CURPH CURRASP DC-O DO-L ENH4 ENO2 ENO2-L
ENO3 ENO3-L EPO4 FM FM-L FM-N FR-O HOLD INH4
INH4-H IPH IPH-H IPH-L IR IR-N ITEMP ITEMP-H ITEMP-L
MAX-RAS MCRT MLSS-L MLSS-S NH4-NPDES NH4EQ NR-L
OPTQW OPTRAS OPTRASP ORGLOAD ORGLOAD-H OUR-L
PADD-L PH-H PH-L PHENOL PHENOL-N PO4-L PO4OUT Q
Q-N QAIR-L QR QRASP-L QW QWAS-N R-DIV1 R-DIV2 R-
DIV3 R-DIV4 R-DIV5 R-DIV6 RC RVSS SA-D SBD SBD-H
SEARCH SSP STATE SUCTION-H SULFIDE SULFIDE-H
TEMP TEMP-H TEMP-L V X XE XE-N XW XWAS-L YAZ )
ASPX-RULES :: (RULE338)
```

```
=====
ASPX-RULES
=====
```

```
RULE338
=====
SUBJECT :: ASPX-RULES
IF :: (YAZ IS NOTKNOWN)
THEN :: (YAZ AND WRITE-DOS-FILE RESULTS.OUT RC
BREAK SSP)
```

```
=====
Frame :: NITRIFICATION
=====
```

```
IDENTIFIER :: "NITRIFICATION-"
PARENTS :: (ASPX)
DISPLAYRESULTS :: YES
PARMGROUP :: NITRIFICATION-PARMS
RULEGROUPS :: (NITRIFICATION-RULES)
OFFSPRING :: (INITIALAMMONIA SA DISSOLVED-OXYGEN
PH PO4 TEMPERATURE) NITRIFICATION-PARMS ::
(INCREASE-DOINCREASE-MCRTINCREASE-MLSSNR PASS)
NITRIFICATION-RULES :: ()
```

```

=====
Frame :: INITIAL
=====

IDENTIFIER :: "INITIAL-"
PARENTS :: (NITRIFICATION ASPX)
DISPLAYRESULTS :: YES
PARMGROUP :: INITIAL-PARMS
RULEGROUPS :: (INITIAL-RULES)
INITIAL-PARMS :: ()
INITIAL-RULES :: (RULE001 RULE002 RULE003 RULE004
RULE005)

```

```

=====
INITIAL-RULES
=====

```

```

RULE001
=====
SUBJECT :: INITIAL-RULES
IF :: ()
THEN :: (READ-FROM-FILE "INPUT" Q CODIN CODOUT
XE X RVSS QW XW QR SBD V TEMP ITEMP PO4OUT
CURDO CURPH IPH CALC-PH INH4 NH4EQ ENH4 ENO2
ENO3 MCRT FM ORGLOAD EPO4 OPTQW OPTRAS OPTRASP
CURRASP PHENOL BOD5/TKN IR SULFIDE )

```

```

RULE002
=====
SUBJECT :: INITIAL-RULES
IF :: (NH4-NPDES IS NOTKNOWN)
THEN :: (READ-FROM-FILE "FUZZY" IPH-L IPH-H
ORGLOAD-H INH4-H ITEMP-L ITEMP-H SULFIDE-H PH-H
PH-L DO-L MLSS-L MLSS-S FM-L FM-N SA-D OUR-L PADD-
L CAUSTIC-H CAUSTIC-L TEMP-L TEMP-H QAIR-L
SUCTION-H BEP-L FR-O DC-O QWAS-N XWAS-L QRASP-L
SBD-H PO4-L ENO2-L ENO3-L NH4-NPDES NR-L PHENOL-N
C/N-H XE-N Q-N IR-N MAX-RAS )

```

```

RULE003
=====
SUBJECT :: INITIAL-RULES
IF :: ()
THEN :: (READ-DOS-FILE "STATE.VAL" STATE AND
READ-DOS-FILE "SEARCH.TCH" SEARCH PASS )

```

```

RULE004
=====
SUBJECT :: INITIAL-RULES
ANTECEDENT :: YES
IF :: (STATE = Severe Problem)
THEN :: (PASS AND INCREASE-DO AND INCREASE-MCRT
AND INCREASE-MLSS)

```

```

RULE005
=====
SUBJECT :: INITIAL-RULES
IF :: (NR IS NOTKNOWN)
THEN :: (NR = (((NH4EQ - ENH4) * (Q + QR)) / ((X *
RVSS) * V)))

```

```

=====
Frame :: AMMONIA
=====

IDENTIFIER :: AMMONIA-
PARENTS :: (NITRIFICATION ASPX)
GOALS :: (R-ENH4)
PARMGROUP :: AMMONIA-PARMS
RULEGROUPS :: (AMMONIA-RULES)
OFFSPRING :: (SUB-NH4)
AMMONIA-PARMS :: (R-ENH4 R-ENO2 R-ENO3)
AMMONIA-RULES :: (RULE032 RULE033 RULE034 RULE035
RULE036 RULE037 RULE038 RULE039 RULE040 RULE041
RULE042 RULE043 RULE044 RULE045 RULE046 RULE047
RULE048 RULE049 )

```

```

=====
AMMONIA-RULES
=====

```

```

RULE032
=====
SUBJECT :: AMMONIA-RULES
UTILITY :: 86
IF :: (R-ENH4 = High)
THEN :: (RC = 14 CF 95)

```

```

RULE033
=====
SUBJECT :: AMMONIA-RULES
IF :: (ENH4 > (NH4-NPDES * 0.75))
THEN :: (R-ENH4 = High CF (CERTAINTY ENH4))

```

```

RULE034
=====
SUBJECT :: AMMONIA-RULES
UTILITY :: 85
IF :: (R-ENH4 = UK)
THEN :: (RC = 15)

```

```

RULE035
=====
SUBJECT :: AMMONIA-RULES
IF :: (ENH4 = UNKNOWN)
THEN :: (R-ENH4 = UK)

```

```

RULE036
=====
SUBJECT :: AMMONIA-RULES
UTILITY :: 84
IF :: (R-ENH4 = Ok AND R-ENO2 = High)
THEN :: (RC = 16 CF 98)

```

```

RULE037
=====
SUBJECT :: AMMONIA-RULES
IF :: (R-ENO3 = Low AND ENO2 >= ENO2-L)
THEN :: (R-ENO2 = High CF (CERTAINTY ENO2))

```

```

RULE038
=====
SUBJECT :: AMMONIA-RULES
UTILITY :: 83
IF :: (R-ENH4 = Ok AND R-ENO2 = UK)
THEN :: (RC = 17)

```

RULE039
 =====
 SUBJECT :: AMMONIA-RULES
 IF :: (R-ENO3 = Low AND ENO2 = UNKNOWN)
 THEN :: (R-ENO2 = UK)

RULE040
 =====
 SUBJECT :: AMMONIA-RULES
 UTILITY :: 82
 IF :: (R-ENH4 = Ok AND R-ENO2 = Ok)
 THEN :: (RC = 18 CF 95)

RULE041
 =====
 SUBJECT :: AMMONIA-RULES
 IF :: (R-ENO3 = Low AND ENO2 < ENO2-L)
 THEN :: (R-ENO2 = Ok CF (CERTAINTY ENO2))

RULE042
 =====
 SUBJECT :: AMMONIA-RULES
 UTILITY :: 81
 IF :: (R-ENH4 = Ok AND R-ENO3 = Ok)
 THEN :: (RC = 19)

RULE043
 =====
 SUBJECT :: AMMONIA-RULES
 IF :: (R-ENH4 = Ok AND ENO3 >= ENO3-L)
 THEN :: (R-ENO3 = Ok CF (CERTAINTY ENO3))

RULE044
 =====
 SUBJECT :: AMMONIA-RULES
 UTILITY :: 80
 IF :: (R-ENH4 = Ok AND R-ENO3 = UK)
 THEN :: (RC = 20)

RULE045
 =====
 SUBJECT :: AMMONIA-RULES
 IF :: (R-ENH4 = Ok AND ENO3 = UNKNOWN)
 THEN :: (R-ENO3 = UK)

RULE046
 =====
 SUBJECT :: AMMONIA-RULES
 UTILITY :: 79
 IF :: (R-ENH4 = Ok AND R-ENO3 = Low)
 THEN :: (RC = 21 CF 95)

RULE047
 =====
 SUBJECT :: AMMONIA-RULES
 IF :: (R-ENH4 = Ok AND ENO3 < ENO3-L)
 THEN :: (R-ENO3 = Low CF (CERTAINTY ENO3))

RULE048
 =====
 SUBJECT :: AMMONIA-RULES
 UTILITY :: 78
 IF :: (R-ENH4 = Ok)
 THEN :: (RC = 22 CF 80)

RULE049
 =====
 SUBJECT :: AMMONIA-RULES
 IF :: (ENH4 <= (NH4-NPDES * 0.75))
 THEN :: (R-ENH4 = Ok CF (CERTAINTY ENH4))

=====
 Frame :: SUB-NH4
 =====
 IDENTIFIER :: "SUB-NH4-"
 PARENTS :: (AMMONIA NITRIFICATION ASPX)
 PROMPT1ST :: PREMISE
 PREMISE :: (\$AND
 (\$OR
 (SAME FRAME PASS)
 (SAME FRAME R-ENH4 "High")))
 PARMGROUP :: SUB-NH4-PARMS
 RULEGROUPS :: (SUB-NH4-RULES)
 SUB-NH4-PARMS :: (R-C/N R-INH4 R-NR)
 SUB-NH4-RULES :: (RULE006 RULE007 RULE008 RULE009
 RULE010 RULE011 RULE012 RULE013 RULE014 RULE015
 RULE016 RULE017 RULE018 RULE019 RULE020 RULE021
 RULE022 RULE023 RULE024 RULE025 RULE026 RULE027
 RULE028 RULE029 RULE030 RULE031)

=====
 SUB-NH4-RULES
 =====

RULE006
 =====
 SUBJECT :: SUB-NH4-RULES
 UTILITY :: 99
 IF :: (R-DIV1 = Yes)
 THEN :: (RC = 1)

RULE007
 =====
 SUBJECT :: SUB-NH4-RULES
 UTILITY :: 97
 IF :: ((PASS OR R-INH4 = High OR R-C/N = High OR R-
 NR = Ok) AND R-DIV1 IS NOTKNOWN AND HOLD = "Yes")
 THEN :: (R-DIV1 = Yes)

RULE008
 =====
 SUBJECT :: SUB-NH4-RULES
 UTILITY :: 98
 IF :: (R-DIV1 = No)
 THEN :: (RC = 2 CF 95 AND INCREASE-DO AND
 INCREASE-MCRT AND INCREASE-MLSS)

RULE009
 =====
 SUBJECT :: SUB-NH4-RULES
 UTILITY :: 98
 IF :: ((PASS OR R-INH4 = High OR R-C/N = High OR R-
 NR = Ok) AND R-DIV1 IS NOTKNOWN AND HOLD = "No")
 THEN :: (R-DIV1 = No)

RULE010
 =====
 SUBJECT :: SUB-NH4-RULES
 UTILITY :: 97
 IF :: (R-DIV1 = UK)
 THEN :: (RC = 3)

RULE011
 =====
 SUBJECT :: SUB-NH4-RULES
 UTILITY :: 99
 IF :: ((PASS OR R-INH4 = High OR R-C/N = High OR R-
 NR = Ok) AND R-DIV1 IS NOTKNOWN AND HOLD = "Not
 Sure") THEN :: (R-DIV1 = UK)

```

RULE012
=====
SUBJECT :: SUB-NH4-RULES
UTILITY :: 96
IF :: (R-DIV1 = N/A)
THEN :: (RC = 4)

RULE013
=====
SUBJECT :: SUB-NH4-RULES
UTILITY :: 100
IF :: ((PASS OR R-INH4 = High OR R-C/N = High OR R-
NR = Ok) AND SEARCH = "Limited" )
THEN :: (R-DIV1 = N/A)

RULE014
=====
SUBJECT :: SUB-NH4-RULES
UTILITY :: 95
IF :: (R-INH4 = Ok)
THEN :: (RC = 5 CF 90)

RULE015
=====
SUBJECT :: SUB-NH4-RULES
IF :: ((PASS OR R-C/N = Ok) AND INH4 < INH4-H)
THEN :: (R-INH4 = Ok CF (CERTAINTY INH4))

RULE016
=====
SUBJECT :: SUB-NH4-RULES
UTILITY :: 94
IF :: (R-INH4 = UK)
THEN :: (RC = 6)

RULE017
=====
SUBJECT :: SUB-NH4-RULES
IF :: ((PASS OR R-C/N = Ok) AND INH4 = UNKNOWN)
THEN :: (R-INH4 = UK)

RULE018
=====
SUBJECT :: SUB-NH4-RULES
UTILITY :: 93
IF :: (R-INH4 = High)
THEN :: (RC = 7 CF 98)

RULE019
=====
SUBJECT :: SUB-NH4-RULES
IF :: ((PASS OR R-C/N = Ok) AND INH4 >= INH4-H)
THEN :: (R-INH4 = High CF (CERTAINTY INH4))

RULE020
=====
SUBJECT :: SUB-NH4-RULES
UTILITY :: 92
IF :: (R-C/N = Ok)
THEN :: (RC = 8 CF 95)

RULE021
=====
SUBJECT :: SUB-NH4-RULES
IF :: ((PASS OR R-NR = Low) AND BOD5/TKN < C/N-H)
THEN :: (R-C/N = Ok CF (CERTAINTY BOD5/TKN))

RULE022
=====
SUBJECT :: SUB-NH4-RULES
UTILITY :: 91
IF :: (R-C/N = UK)
THEN :: (RC = 9)

RULE023
=====
SUBJECT :: SUB-NH4-RULES
IF :: ((PASS OR R-NR = Low) AND BOD5/TKN =
UNKNOWN)
THEN :: (R-C/N = UK)

RULE024
=====
SUBJECT :: SUB-NH4-RULES
UTILITY :: 90
IF :: (R-C/N = High)
THEN :: (RC = 10 CF 98)

RULE025
=====
SUBJECT :: SUB-NH4-RULES
IF :: ((PASS OR R-NR = Low) AND BOD5/TKN >= C/N-
H) THEN :: (R-C/N = High CF (CERTAINTY BOD5/TKN))

RULE026
=====
SUBJECT :: SUB-NH4-RULES
UTILITY :: 89
IF :: (R-NR = Low)
THEN :: (RC = 11 CF 80)

RULE027
=====
SUBJECT :: SUB-NH4-RULES
IF :: ((PASS OR R-ENH4 = High) AND NR < NR-L)
THEN :: (R-NR = Low CF (CERTAINTY ENH4))

RULE028
=====
SUBJECT :: SUB-NH4-RULES
UTILITY :: 88
IF :: (R-NR = UK)
THEN :: (RC = 12)

RULE029
=====
SUBJECT :: SUB-NH4-RULES
IF :: ((PASS OR R-ENH4 = High) AND NR = Cannot
Calculate) THEN :: (R-NR = UK)

RULE030
=====
SUBJECT :: SUB-NH4-RULES
UTILITY :: 87
IF :: (R-NR = Ok)
THEN :: (RC = 13 CF 95)

RULE031
=====
SUBJECT :: SUB-NH4-RULES
IF :: ((PASS OR R-ENH4 = High) AND NR >= NR-L)
THEN :: (R-NR = Ok CF (CERTAINTY ENH4))

```

```

=====
Frame :: SA
=====

IDENTIFIER :: "SA-"
PARENTS :: (NITRIFICATION ASPX)
GOALS :: (R-SA)
PARMGROUP :: SA-PARMS
RULEGROUPS :: (SA-RULES)
OFFSPRING :: (SUB-SA)
SA-PARMS :: (R-SA)
SA-RULES :: (RULE147 RULE148 RULE149 RULE150
RULE151 RULE152 RULE153)

```

```

=====
SA-RULES
=====

```

RULE147

```

=====
SUBJECT :: SA-RULES
UTILITY :: 28
IF :: (R-SA = Low)
THEN :: (RC = 72 CF 98)

```

RULE148

```

=====
SUBJECT :: SA-RULES
IF :: (MCRT < SA-D)
THEN :: (R-SA = Low CF (CERTAINTY MCRT))

```

RULE149

```

=====
SUBJECT :: SA-RULES
UTILITY :: 27
IF :: (R-SA = UK)
THEN :: (RC = 73)

```

RULE150

```

=====
SUBJECT :: SA-RULES
IF :: (MCRT = Cannot Calculate)
THEN :: (R-SA = UK)

```

RULE151

```

=====
SUBJECT :: SA-RULES
UTILITY :: 26
IF :: (R-SA = Ok AND INCREASE-MCRT)
THEN :: (RC = 74)

```

RULE152

```

=====
SUBJECT :: SA-RULES
UTILITY :: 25
IF :: (R-SA = Ok AND 1 INCREASE-MCRT)
THEN :: (RC = 75)

```

RULE153

```

=====
SUBJECT :: SA-RULES
UTILITY :: 24
IF :: (MCRT >= SA-D)
THEN :: (R-SA = Ok CF (CERTAINTY MCRT) AND RC =
76 CF 95)

```

```

=====
Frame :: SUB-SA
=====

```

IDENTIFIER :: "SUB-SA-"

```

PARENTS :: (SA NITRIFICATION ASPX)
PROMPT1ST :: PREMISE
PREMISE :: ($AND
($OR
(SAME FRAME PASS)
(SAME FRAME R-SA "Low")
(SAME FRAME INCREASE-MCRT)))
PARMGROUP :: SUB-SA-PARMS
RULEGROUPS :: (SUB-SA-RULES)
SUB-SA-PARMS :: (OUR R-FM R-OUR R-PHL R-Q R-RAS R-
SB R-WAS R-X R-X2 R-XE R-XW)
SUB-SA-RULES :: (RULE050 RULE051 RULE052 RULE053
RULE054 RULE055 RULE056 RULE057 RULE058 RULE059
RULE060 RULE061 RULE062 RULE063 RULE064 RULE065
RULE066 RULE067 RULE068 RULE069 RULE070 RULE071
RULE072 RULE073 RULE074 RULE075 RULE076 RULE077
RULE078 RULE079 RULE080 RULE081 RULE082 RULE083
RULE084 RULE085 RULE086 RULE087 RULE088 RULE089
RULE090 RULE091 RULE092 RULE093 RULE094 RULE095
RULE096 RULE097 RULE098 RULE099 RULE100 RULE101
RULE102 RULE103 RULE104 RULE105 RULE106 RULE107
RULE108 RULE109 RULE110 RULE111 RULE112 RULE113
RULE114 RULE115 RULE116 RULE117 RULE118 RULE119
RULE120 RULE121 RULE122 RULE123 RULE124 RULE125
RULE126 RULE127 RULE128 RULE129 RULE130 RULE131
RULE132 RULE133 RULE134 RULE135 RULE136 RULE137
RULE138 RULE139 RULE140 RULE141 RULE142 RULE143
RULE144 RULE145 RULE146)

```

```

=====
SUB-SA-RULES
=====

```

RULE050

```

=====
SUBJECT :: SUB-SA-RULES
UTILITY :: 77
IF :: (R-DIV2 = No)
THEN :: (RC = 23)

```

RULE051

```

=====
SUBJECT :: SUB-SA-RULES
UTILITY :: 93
IF :: (R-Q = High AND R-DIV2 IS NOTKNOWN AND
HOLD = No) THEN :: (R-DIV2 = No)

```

RULE052

```

=====
SUBJECT :: SUB-SA-RULES
UTILITY :: 76
IF :: (R-DIV2 = Yes)
THEN :: (RC = 24)

```

RULE053

```

=====
SUBJECT :: SUB-SA-RULES
UTILITY :: 94
IF :: (R-Q = High AND R-DIV2 IS NOTKNOWN AND
HOLD = Yes) THEN :: (R-DIV2 = Yes)

```

```

RULE054
=====
SUBJECT :: SUB-SA-RULES
UTILITY :: 75
IF :: (R-DIV2 = UK)
THEN :: (RC = 25)

RULE055
=====
SUBJECT :: SUB-SA-RULES
UTILITY :: 95
IF :: (R-Q = High AND R-DIV2 IS NOTKNOWN AND
HOLD = Not Sure) THEN :: (R-DIV2 = UK)

RULE056
=====
SUBJECT :: SUB-SA-RULES
UTILITY :: 74
IF :: (R-DIV2 = N/A)
THEN :: (RC = 26)

RULE057
=====
SUBJECT :: SUB-SA-RULES
UTILITY :: 96
IF :: (R-Q = High AND SEARCH = Limited)
THEN :: (R-DIV2 = N/A)

RULE058
=====
SUBJECT :: SUB-SA-RULES
UTILITY :: 73
IF :: (R-Q = High)
THEN :: (RC = 27 CF 95)

RULE059
=====
SUBJECT :: SUB-SA-RULES
IF :: (R-XE = Ok AND Q > Q-N)
THEN :: (R-Q = High CF (CERTAINTY Q))

RULE060
=====
SUBJECT :: SUB-SA-RULES
UTILITY :: 72
IF :: (R-Q = UK)
THEN :: (RC = 28)

RULE061
=====
SUBJECT :: SUB-SA-RULES
IF :: (R-XE = Ok AND Q = UNKNOWN)
THEN :: (R-Q = UK)

RULE062
=====
SUBJECT :: SUB-SA-RULES
UTILITY :: 71
IF :: (R-Q = Ok)
THEN :: (RC = 29 CF 90)

RULE063
=====
SUBJECT :: SUB-SA-RULES
IF :: (R-XE = Ok AND Q <= Q-N)
THEN :: (R-Q = Ok CF (CERTAINTY Q))

RULE064
=====
SUBJECT :: SUB-SA-RULES
UTILITY :: 70
IF :: (R-XE = High)
THEN :: (RC = 30 CF 90 AND CHECK-SSP)

RULE065
=====
SUBJECT :: SUB-SA-RULES
IF :: ((R-OUR = Ok OR R-DIV3 = No) AND XE > XE-N)
THEN :: (R-XE = High CF (CERTAINTY XE))

RULE066
=====
SUBJECT :: SUB-SA-RULES
UTILITY :: 69
IF :: (R-XE = UK)
THEN :: (RC = 31)

RULE067
=====
SUBJECT :: SUB-SA-RULES
IF :: ((R-OUR = Ok OR R-DIV3 = No) AND XE =
UNKNOWN) THEN :: (R-XE = UK)

RULE068
=====
SUBJECT :: SUB-SA-RULES
UTILITY :: 68
IF :: (R-XE = Ok)
THEN :: (RC = 32 CF 95)

RULE069
=====
SUBJECT :: SUB-SA-RULES
IF :: ((R-OUR = Ok OR R-DIV3 = No) AND XE <= XE-N)
THEN :: (R-XE = Ok CF (CERTAINTY XE))

RULE070
=====
SUBJECT :: SUB-SA-RULES
UTILITY :: 67
IF :: (R-DIV3 = No)
THEN :: (RC = 33)

RULE071
=====
SUBJECT :: SUB-SA-RULES
UTILITY :: 89
IF :: (R-RAS = Max AND R-DIV3 IS NOTKNOWN AND
HOLD = No) THEN :: (R-DIV3 = No)

RULE072
=====
SUBJECT :: SUB-SA-RULES
UTILITY :: 66
IF :: (R-DIV3 = Yes)
THEN :: (RC = 34)

RULE073
=====
SUBJECT :: SUB-SA-RULES
UTILITY :: 90
IF :: (R-RAS = Max AND R-DIV3 IS NOTKNOWN AND
HOLD = Yes) THEN :: (R-DIV3 = Yes)

```



```

RULE074
=====
SUBJECT :: SUB-SA-RULES
UTILITY :: 65
IF :: (R-DIV3 = UK)
THEN :: (RC = 35)

RULE075
=====
SUBJECT :: SUB-SA-RULES
UTILITY :: 91
IF :: (R-RAS = Max AND R-DIV3 IS NOTKNOWN AND
HOLD = Not Sure) THEN :: (R-DIV3 = UK)

RULE076
=====
SUBJECT :: SUB-SA-RULES
UTILITY :: 64
IF :: (R-DIV3 = N/A)
THEN :: (RC = 36)

RULE077
=====
SUBJECT :: SUB-SA-RULES
UTILITY :: 92
IF :: (R-RAS = Max AND SEARCH = Limited)
THEN :: (R-DIV3 = N/A)

RULE078
=====
SUBJECT :: SUB-SA-RULES
UTILITY :: 63
IF :: (R-DIV4 = No)
THEN :: (RC = 37 AND RC = 41)

RULE079
=====
SUBJECT :: SUB-SA-RULES
UTILITY :: 86
IF :: ((R-PHL = High OR R-OUR = Low) AND R-DIV4 IS
NOTKNOWN AND HOLD = "No" )
THEN :: (R-DIV4 = No)

RULE080
=====
SUBJECT :: SUB-SA-RULES
UTILITY :: 62
IF :: (R-DIV4 = Yes)
THEN :: (RC = 38)

RULE081
=====
SUBJECT :: SUB-SA-RULES
UTILITY :: 88
IF :: ((R-PHL = High OR R-OUR = Low) AND R-DIV4 IS
NOTKNOWN AND HOLD = "Yes" )
THEN :: (R-DIV4 = Yes)

RULE082
=====
SUBJECT :: SUB-SA-RULES
UTILITY :: 61
IF :: (R-DIV4 = UK)
THEN :: (RC = 39)

RULE083
=====
SUBJECT :: SUB-SA-RULES
UTILITY :: 87

IF :: ((R-PHL = High OR R-OUR = Low) AND R-DIV4 IS
NOTKNOWN AND HOLD = "Not Sure" )
THEN :: (R-DIV4 = UK)

RULE084
=====
SUBJECT :: SUB-SA-RULES
UTILITY :: 60
IF :: (R-DIV4 = N/A)
THEN :: (RC = 40)

RULE085
=====
SUBJECT :: SUB-SA-RULES
UTILITY :: 88
IF :: ((R-PHL = High OR R-OUR = Low) AND SEARCH =
Limited) THEN :: (R-DIV4 = N/A)

RULE086
=====
SUBJECT :: SUB-SA-RULES
UTILITY :: 58
IF :: (R-PHL = Ok)
THEN :: (RC = 42 CF 90 AND RC = 41 CF 90)

RULE087
=====
SUBJECT :: SUB-SA-RULES
IF :: ((PASS OR R-X = Low) AND PHENOL < = PHENOL-
N) THEN :: (R-PHL = Ok CF (CERTAINTY PHENOL))

RULE088
=====
SUBJECT :: SUB-SA-RULES
UTILITY :: 57
IF :: (R-PHL = High)
THEN :: (RC = 43 CF 98)

RULE089
=====
SUBJECT :: SUB-SA-RULES
IF :: ((PASS OR R-X = Low) AND PHENOL > PHENOL-N)
THEN :: (R-PHL = High CF (CERTAINTY PHENOL))

RULE090
=====
SUBJECT :: SUB-SA-RULES
UTILITY :: 56
IF :: (R-PHL = UK)
THEN :: (RC = 44)

RULE091
=====
SUBJECT :: SUB-SA-RULES
IF :: ((PASS OR R-X = Low) AND PHENOL = UNKNOWN)
THEN :: (R-PHL = N/A)

RULE092
=====
SUBJECT :: SUB-SA-RULES
UTILITY :: 55
IF :: (R-OUR = Low)
THEN :: (RC = 45 CF 95)

RULE093
=====
SUBJECT :: SUB-SA-RULES
UTILITY :: 97

```

IF :: ((PASS OR R-X2 = Ok) AND R-OUR IS NOTKNOWN
AND OUR <= OUR-L)
THEN :: (R-OUR = Low)

RULE094

=====

SUBJECT :: SUB-SA-RULES
UTILITY :: 54
IF :: (R-OUR = Ok)
THEN :: (RC = 46 CF 95)

RULE095

=====

SUBJECT :: SUB-SA-RULES
UTILITY :: 98
IF :: ((PASS OR R-X2 = Ok) AND R-OUR IS NOTKNOWN
AND OUR > OUR-L) THEN :: (R-OUR = Ok)

RULE096

=====

SUBJECT :: SUB-SA-RULES
UTILITY :: 53
IF :: (R-OUR = UK)
THEN :: (RC = 47)

RULE097

=====

SUBJECT :: SUB-SA-RULES
UTILITY :: 99
IF :: ((PASS OR R-X2 = Ok) AND R-OUR IS NOTKNOWN
AND OUR = 999) THEN :: (R-OUR = UK)

RULE098

=====

SUBJECT :: SUB-SA-RULES
UTILITY :: 52
IF :: (R-OUR = N/A)
THEN :: (RC = 48)

RULE099

=====

SUBJECT :: SUB-SA-RULES
UTILITY :: 100
IF :: ((PASS OR R-X2 = Ok) AND SEARCH = Limited)
THEN :: (R-OUR = N/A)

RULE100

=====

SUBJECT :: SUB-SA-RULES
UTILITY :: 51
IF :: (R-X = Low)
THEN :: (RC = 49 CF 95)

RULE101

=====

SUBJECT :: SUB-SA-RULES
IF :: (R-FM = High AND X <= MLSS-L)
THEN :: (R-X = Low CF (CERTAINTY X))

RULE102

=====

SUBJECT :: SUB-SA-RULES
UTILITY :: 50
IF :: (R-X = UK)
THEN :: (RC = 50)

RULE103

=====

SUBJECT :: SUB-SA-RULES
IF :: (R-FM = High AND X = UNKNOWN)
THEN :: (R-X = UK)

RULE104

=====

SUBJECT :: SUB-SA-RULES
UTILITY :: 49
IF :: (R-X = Ok OR R-X2 = Low)
THEN :: (RC = 51)

RULE105

=====

SUBJECT :: SUB-SA-RULES
IF :: (R-FM = High AND X > MLSS-L)
THEN :: (R-X = Ok CF (CERTAINTY X))

RULE106

=====

SUBJECT :: SUB-SA-RULES
IF :: (R-FM = Ok AND X <= MLSS-L)
THEN :: (R-X2 = Low CF (CERTAINTY X))

RULE107

=====

SUBJECT :: SUB-SA-RULES
UTILITY :: 48
IF :: (R-X2 = UK)
THEN :: (RC = 52)

RULE108

=====

SUBJECT :: SUB-SA-RULES
IF :: (R-FM = Ok AND X = UNKNOWN)
THEN :: (R-X2 = UK)

RULE109

=====

SUBJECT :: SUB-SA-RULES
UTILITY :: 47
IF :: (R-X2 = Ok)
THEN :: (RC = 53 CF 95)

RULE110

=====

SUBJECT :: SUB-SA-RULES
IF :: (R-FM = Ok AND X > MLSS-L)
THEN :: (R-X2 = Ok CF (CERTAINTY X))

RULE111

=====

SUBJECT :: SUB-SA-RULES
UTILITY :: 46
IF :: (R-RAS = Low)
THEN :: (RC = 54 CF 98)

RULE112

=====

SUBJECT :: SUB-SA-RULES
IF :: ((PASS OR R-SB = High) AND (QR < OPTRAS OR
(QR / Q) < (QRASP-L / 100)))
THEN :: (R-RAS = Low CF (CERTAINTY QR))

RULE113

=====

SUBJECT :: SUB-SA-RULES
UTILITY :: 45
IF :: (R-RAS = Max)
THEN :: (RC = 55 CF 95)

RULE114

=====

SUBJECT :: SUB-SA-RULES
UTILITY :: 100

IF :: ((PASS OR R-SB = High) AND QR >= MAX-RAS)
THEN :: (R-RAS = Max CF (CERTAINTY QR))

RULE115
=====

SUBJECT :: SUB-SA-RULES
UTILITY :: 44
IF :: (R-RAS = UK)
THEN :: (RC = 56)

RULE116
=====

SUBJECT :: SUB-SA-RULES
IF :: ((PASS OR R-SB = High) AND QR = UNKNOWN)
THEN :: (R-RAS = UK)

RULE117
=====

SUBJECT :: SUB-SA-RULES
UTILITY :: 43
IF :: (R-RAS = Ok)
THEN :: (RC = 57 CF 90)

RULE118
=====

SUBJECT :: SUB-SA-RULES
IF :: ((PASS OR R-SB = High) AND QR >= OPTRAS AND
(QR / Q) >= (QRASP-L / 100))
THEN :: (R-RAS = Ok CF (CERTAINTY QR))

RULE119
=====

SUBJECT :: SUB-SA-RULES
UTILITY :: 42
IF :: (R-SB = Ok)
THEN :: (RC = 58 CF 95)

RULE120
=====

SUBJECT :: SUB-SA-RULES
IF :: ((R-WAS = High OR R-WAS = Norm) AND SBD <
SBD-H) THEN :: (R-SB = Ok CF (CERTAINTY SBD))

RULE121
=====

SUBJECT :: SUB-SA-RULES
UTILITY :: 41
IF :: (R-SB = High)
THEN :: (RC = 59 CF 90)

RULE122
=====

SUBJECT :: SUB-SA-RULES
IF :: ((R-WAS = High OR R-WAS = Norm) AND SBD >=
SBD-H) THEN :: (R-SB = High CF (CERTAINTY SBD))

RULE123
=====

SUBJECT :: SUB-SA-RULES
UTILITY :: 40
IF :: (R-SB = UK)
THEN :: (RC = 60)

RULE124
=====

SUBJECT :: SUB-SA-RULES
IF :: ((R-WAS = High OR R-WAS = Norm) AND SBD =
UNKNOWN) THEN :: (R-SB = UK)

RULE125
=====

SUBJECT :: SUB-SA-RULES
UTILITY :: 39
IF :: (R-WAS = High)
THEN :: (RC = 61 CF 98)

RULE126
=====

SUBJECT :: SUB-SA-RULES
IF :: ((PASS OR R-SA = Low OR INCREASE-MCRT) AND
QW > (1.1 * OPTQW)) THEN :: (R-WAS = High CF
(CERTAINTY QW))

RULE127
=====

SUBJECT :: SUB-SA-RULES
UTILITY :: 38
IF :: (R-WAS = Norm)
THEN :: (RC = 62 CF 95)

RULE128
=====

SUBJECT :: SUB-SA-RULES
IF :: ((PASS OR R-SA = Low OR INCREASE-MCRT) AND
QW >= (0.9 * OPTQW) AND QW <= (1.1 * OPTQW))
THEN :: (R-WAS = Norm CF (CERTAINTY QW))

RULE129
=====

SUBJECT :: SUB-SA-RULES
UTILITY :: 37
IF :: (R-WAS = Low)
THEN :: (RC = 69 CF 95)

RULE130
=====

SUBJECT :: SUB-SA-RULES
IF :: ((PASS OR R-SA = Low OR INCREASE-MCRT) AND
QW > 0 AND QW <= (0.9 * OPTQW))
THEN :: (R-WAS = Low CF (CERTAINTY QW))

RULE131
=====

SUBJECT :: SUB-SA-RULES
UTILITY :: 36
IF :: (R-WAS = UK)
THEN :: (RC = 70)

RULE132
=====

SUBJECT :: SUB-SA-RULES
IF :: ((PASS OR R-SA = Low OR INCREASE-MCRT) AND
QW = UNKNOWN) THEN :: (R-WAS = UK)

RULE133
=====

SUBJECT :: SUB-SA-RULES
UTILITY :: 35
IF :: (R-WAS = Zero)
THEN :: (RC = 71)

RULE134
=====

SUBJECT :: SUB-SA-RULES
IF :: ((PASS OR R-SA = Low OR INCREASE-MCRT) AND
QW = 0) THEN :: (R-WAS = Zero CF (CERTAINTY QW))

```

RULE135
=====
SUBJECT :: SUB-SA-RULES
UTILITY :: 34
IF :: (R-FM = High)
THEN :: (RC = 63 CF 90)

RULE136
=====
SUBJECT :: SUB-SA-RULES
IF :: ((PASS OR R-XW = Ok) AND FM > FM-N)
THEN :: (R-FM = High CF (CERTAINTY FM))

RULE137
=====
SUBJECT :: SUB-SA-RULES
UTILITY :: 33
IF :: (R-FM = Ok)
THEN :: (RC = 64 CF 95)

RULE138
=====
SUBJECT :: SUB-SA-RULES
IF :: ((PASS OR R-XW = Ok) AND FM <= FM-N)
THEN :: (R-FM = Ok CF (CERTAINTY FM))

RULE139
=====
SUBJECT :: SUB-SA-RULES
UTILITY :: 32
IF :: (R-FM = UK)
THEN :: (RC = 65)

RULE140
=====
SUBJECT :: SUB-SA-RULES
IF :: ((PASS OR R-XW = Ok) AND FM = Cannot Calculate)
THEN :: (R-FM = UK)

RULE141
=====
SUBJECT :: SUB-SA-RULES
UTILITY :: 31
IF :: (R-XW = Low)
THEN :: (RC = 66 CF 80 AND CHECK-SSP)

RULE142
=====
SUBJECT :: SUB-SA-RULES
IF :: ((PASS OR R-WAS = Low OR R-WAS = Zero OR R-
RAS = Ok) AND XW <= XWAS-L)
THEN :: (R-XW = Low CF (CERTAINTY XW))

RULE143
=====
SUBJECT :: SUB-SA-RULES
UTILITY :: 30
IF :: (R-XW = Ok)
THEN :: (RC = 67 CF 95)

RULE144
=====
SUBJECT :: SUB-SA-RULES
IF :: ((PASS OR R-WAS = Low OR R-WAS = Zero OR R-
RAS = Ok) AND XW > XWAS-L)
THEN :: (R-XW = Ok CF (CERTAINTY XW))

RULE145
=====
SUBJECT :: SUB-SA-RULES
UTILITY :: 29
IF :: (R-XW = UK)
THEN :: (RC = 68)

RULE146
=====
SUBJECT :: SUB-SA-RULES
IF :: ((PASS OR R-WAS = Low OR R-WAS = Zero OR R-
RAS = Ok) AND XW = UNKNOWN)
THEN :: (R-XW = UK)

=====
Frame :: DISSOLVED-OXYGEN
=====

IDENTIFIER :: "DISSOLVED-OXYGEN-"
PARENTS :: (NITRIFICATION ASPX)
GOALS :: (R-DO)
PARMGROUP :: DISSOLVED-OXYGEN-PARMS
RULEGROUPS :: (DISSOLVED-OXYGEN-RULES)
OFFSPRING :: (SUB-DO)
DISSOLVED-OXYGEN-PARMS :: (MBA R-DO R-MBA)
DISSOLVED-OXYGEN-RULES :: (RULE246 RULE247
RULE248 RULE249 RULE250 RULE251 RULE252 RULE253)

=====
DISSOLVED-OXYGEN-RULES
=====

RULE246
=====
SUBJECT :: DISSOLVED-OXYGEN-RULES
UTILITY :: -23
IF :: (R-DO = Ok AND INCREASE-DO)
THEN :: (RC = 123)

RULE247
=====
SUBJECT :: DISSOLVED-OXYGEN-RULES
IF :: (R-DO = Ok AND ! INCREASE-DO)
THEN :: (RC = 124)

RULE248
=====
SUBJECT :: DISSOLVED-OXYGEN-RULES
UTILITY :: -25
IF :: (R-DO = Ok)
THEN :: (RC = 125 CF 95)

RULE249
=====
SUBJECT :: DISSOLVED-OXYGEN-RULES
IF :: (CURDO >= DO-L)
THEN :: (R-DO = Ok CF (CERTAINTY CURDO))

RULE250
=====
SUBJECT :: DISSOLVED-OXYGEN-RULES
UTILITY :: -26
IF :: (R-DO = UK)
THEN :: (RC = 126)

```

```

RULE251
=====
SUBJECT :: DISSOLVED-OXYGEN-RULES
IF :: (CURDO = UNKNOWN)
THEN :: (R-DO = UK)

RULE252
=====
SUBJECT :: DISSOLVED-OXYGEN-RULES
UTILITY :: -27
IF :: (R-DO = Low)
THEN :: (RC = 127 CF 95)

RULE253
=====
SUBJECT :: DISSOLVED-OXYGEN-RULES
IF :: (CURDO < DO-L)
THEN :: (R-DO = Low CF (CERTAINTY CURDO))

=====
Frame :: SUB-DO
=====

IDENTIFIER :: "SUB-DO-"
PARENTS :: (DISSOLVED-OXYGEN NITRIFICATION ASPX)
PROMPT1ST :: PREMISE
PREMISE :: ($AND
($OR
(SAME FRAME PASS)
(SAME FRAME R-DO "Low")
(SAME FRAME INCREASE-DO)))
PARMGROUP :: SUB-DO-PARMS
RULEGROUPS :: (SUB-DO-RULES)
SUB-DO-PARMS :: (ADIVERT BEP BLOWER DIFFUSER
FILTER QAIR R-ADIV R-BEP R-DC R-FM2 R-FR R-QA R-
SUC R-SURF R-VAL R-X3 SUCTION SURFACE VALVE )
SUB-DO-RULES :: (RULE154 RULE155 RULE156 RULE157
RULE158 RULE159 RULE160 RULE161 RULE162 RULE163
RULE164 RULE165 RULE166 RULE167 RULE168 RULE169
RULE170 RULE171 RULE172 RULE173 RULE174 RULE175
RULE176 RULE177 RULE178 RULE179 RULE180 RULE181
RULE182 RULE183 RULE184 RULE185 RULE186 RULE187
RULE188 RULE189 RULE190 RULE191 RULE192 RULE193
RULE194 RULE195 RULE196 RULE197 RULE198 RULE199
RULE200 RULE201 RULE202 RULE203 RULE204 RULE205
RULE206 RULE207 RULE208 RULE209 RULE210 RULE211
RULE212 RULE213 RULE214 RULE215 RULE216 RULE217
RULE218 RULE219 RULE220 RULE221 RULE222 RULE223
RULE224 RULE225 RULE226 RULE227 RULE228 RULE229
RULE230 RULE231 RULE232 RULE233 RULE234 RULE235
RULE236 RULE237 RULE238 RULE239 RULE240 RULE241
RULE242 RULE243 RULE244 RULE245 )

```

```

=====
SUB-DO-RULES
=====

```

```

RULE154
=====
SUBJECT :: SUB-DO-RULES
UTILITY :: 23
IF :: (R-FR = New)
THEN :: (RC = 77 CF 90)

```

```

RULE155
=====
SUBJECT :: SUB-DO-RULES
UTILITY :: 97
IF :: ((PASS OR R-SUC = High) AND FILTER <= FR-O)
THEN :: (R-FR = New)

```

```

RULE156
=====
SUBJECT :: SUB-DO-RULES
UTILITY :: 22
IF :: (R-FR = Old)
THEN :: (RC = 78 CF 95)

```

```

RULE157
=====
SUBJECT :: SUB-DO-RULES
UTILITY :: 98
IF :: ((PASS OR R-SUC = High) AND FILTER > FR-O)
THEN :: (R-FR = Old)

```

```

RULE158
=====
SUBJECT :: SUB-DO-RULES
UTILITY :: 21
IF :: (R-FR = UK)
THEN :: (RC = 79)

```

```

RULE159
=====
SUBJECT :: SUB-DO-RULES
UTILITY :: 99
IF :: ((PASS OR R-SUC = High) AND FILTER = 999)
THEN :: (R-FR = UK)

```

```

RULE160
=====
SUBJECT :: SUB-DO-RULES
UTILITY :: 20
IF :: (R-FR = N/A)
THEN :: (RC = 80)

```

```

RULE161
=====
SUBJECT :: SUB-DO-RULES
UTILITY :: 100
IF :: ((PASS OR R-SUC = High) AND SEARCH = Limited)
THEN :: (R-FR = N/A)

```

```

RULE162
=====
SUBJECT :: SUB-DO-RULES
UTILITY :: 19
IF :: (R-SUC = Ok)
THEN :: (RC = 81 CF 95)

```

```

RULE163
=====
SUBJECT :: SUB-DO-RULES
UTILITY :: 97
IF :: ((PASS OR R-ADIV = No OR R-BEP = Ok) AND
SUCTION <= SUCTION-H) THEN :: (R-SUC = Ok)

```

```

RULE164
=====
SUBJECT :: SUB-DO-RULES
UTILITY :: 18
IF :: (R-SUC = High)
THEN :: (RC = 82 CF 95)

```

RULE165
 =====
 SUBJECT :: SUB-DO-RULES
 UTILITY :: 98
 IF :: ((PASS OR R-ADIV = No OR R-BEP = Ok) AND
 SUCTION > SUCTION-H) THEN :: (R-SUC = High)

RULE166
 =====
 SUBJECT :: SUB-DO-RULES
 UTILITY :: 17
 IF :: (R-SUC = UK)
 THEN :: (RC = 83)

RULE167
 =====
 SUBJECT :: SUB-DO-RULES
 UTILITY :: 99
 IF :: ((PASS OR R-ADIV = No OR R-BEP = Ok) AND
 SUCTION = 999) THEN :: (R-SUC = UK)

RULE168
 =====
 SUBJECT :: SUB-DO-RULES
 UTILITY :: 16
 IF :: (R-SUC = N/A)
 THEN :: (RC = 84)

RULE169
 =====
 SUBJECT :: SUB-DO-RULES
 UTILITY :: 100
 IF :: ((PASS OR R-ADIV = No OR R-BEP = Ok) AND
 SEARCH = Limited) THEN :: (R-SUC = N/A)

RULE170
 =====
 SUBJECT :: SUB-DO-RULES
 UTILITY :: 15
 IF :: (R-ADIV = Yes)
 THEN :: (RC = 85)

RULE171
 =====
 SUBJECT :: SUB-DO-RULES
 UTILITY :: 97
 IF :: (R-BEP = Low AND ADIVERT = Yes)
 THEN :: (R-ADIV = Yes)

RULE172
 =====
 SUBJECT :: SUB-DO-RULES
 UTILITY :: 14
 IF :: (R-ADIV = UK)
 THEN :: (RC = 86)

RULE173
 =====
 SUBJECT :: SUB-DO-RULES
 UTILITY :: 99
 IF :: (R-BEP = Low AND ADIVERT = 999)
 THEN :: (R-ADIV = UK)

RULE174
 =====
 SUBJECT :: SUB-DO-RULES
 UTILITY :: 13
 IF :: (R-ADIV = N/A)
 THEN :: (RC = 87)

RULE175
 =====
 SUBJECT :: SUB-DO-RULES
 UTILITY :: 100
 IF :: (R-BEP = Low AND SEARCH = Limited)
 THEN :: (R-ADIV = N/A)

RULE176
 =====
 SUBJECT :: SUB-DO-RULES
 UTILITY :: 12
 IF :: (R-ADIV = No)
 THEN :: (RC = 88 CF 75)

RULE177
 =====
 SUBJECT :: SUB-DO-RULES
 UTILITY :: 98
 IF :: (R-BEP = Low AND ADIVERT = No)
 THEN :: (R-ADIV = No)

RULE178
 =====
 SUBJECT :: SUB-DO-RULES
 UTILITY :: 11
 IF :: (R-BEP = Low)
 THEN :: (RC = 89 CF 98)

RULE179
 =====
 SUBJECT :: SUB-DO-RULES
 IF :: ((PASS OR R-VAL = Open OR R-QA = Ok) AND BEP
 < BEP-L) THEN :: (R-BEP = Low)

RULE180
 =====
 SUBJECT :: SUB-DO-RULES
 UTILITY :: 10
 IF :: (R-BEP = Ok)
 THEN :: (RC = 90 CF 90)

RULE181
 =====
 SUBJECT :: SUB-DO-RULES
 IF :: ((PASS OR R-VAL = Open OR R-QA = Ok) AND BEP
 > = BEP-L) THEN :: (R-BEP = Ok)

RULE182
 =====
 SUBJECT :: SUB-DO-RULES
 UTILITY :: 9
 IF :: (R-BEP = UK)
 THEN :: (RC = 91)

RULE183
 =====
 SUBJECT :: SUB-DO-RULES
 IF :: ((PASS OR R-VAL = Open OR R-QA = Ok) AND BEP
 = 999) THEN :: (R-BEP = UK)

RULE184
 =====
 SUBJECT :: SUB-DO-RULES
 UTILITY :: 8
 IF :: (R-BEP = N/A)
 THEN :: (RC = 92)

RULE185
 =====
 SUBJECT :: SUB-DO-RULES
 IF :: ((PASS OR R-VAL = Open OR R-QA = Ok) AND
 SEARCH = Limited) THEN :: (R-BEP = N/A)

RULE186
 =====
 SUBJECT :: SUB-DO-RULES
 UTILITY :: 7
 IF :: (R-VAL = Closed)
 THEN :: (RC = 93)

RULE187
 =====
 SUBJECT :: SUB-DO-RULES
 IF :: (R-QA = Low AND VALVE = Closed)
 THEN :: (R-VAL = Closed)

RULE188
 =====
 SUBJECT :: SUB-DO-RULES
 UTILITY :: 6
 IF :: (R-VAL = UK)
 THEN :: (RC = 94)

RULE189
 =====
 SUBJECT :: SUB-DO-RULES
 IF :: (R-QA = Low AND VALVE = Not Sure)
 THEN :: (R-VAL = UK)

RULE190
 =====
 SUBJECT :: SUB-DO-RULES
 UTILITY :: 5
 IF :: (R-VAL = N/A)
 THEN :: (RC = 95)

RULE191
 =====
 SUBJECT :: SUB-DO-RULES
 UTILITY :: 100
 IF :: (R-QA = Low AND SEARCH = Limited)
 THEN :: (R-VAL = N/A)

RULE192
 =====
 SUBJECT :: SUB-DO-RULES
 UTILITY :: 4
 IF :: (R-VAL = Open)
 THEN :: (RC = 96 CF 70)

RULE193
 =====
 SUBJECT :: SUB-DO-RULES
 IF :: (R-QA = Low AND VALVE = Open)
 THEN :: (R-VAL = Open)

RULE194
 =====
 SUBJECT :: SUB-DO-RULES
 UTILITY :: 3
 IF :: (R-QA = Low)
 THEN :: (RC = 97 CF 98)

RULE195
 =====
 SUBJECT :: SUB-DO-RULES
 UTILITY :: 97
 IF :: ((PASS OR R-SURF = Ok OR R-DC = Ok) AND
 BLOWER IS KNOWN AND QAIR < (BLOWER * QAIR-L))
 THEN :: (R-QA = Low)

RULE196
 =====
 SUBJECT :: SUB-DO-RULES
 UTILITY :: 2
 IF :: (R-QA = Ok)
 THEN :: (RC = 98 CF 90)

RULE197
 =====
 SUBJECT :: SUB-DO-RULES
 UTILITY :: 98
 IF :: ((PASS OR R-SURF = Ok OR R-DC = Ok) AND
 BLOWER IS KNOWN AND QAIR >= (BLOWER * QAIR-L))
 THEN :: (R-QA = Ok)

RULE198
 =====
 SUBJECT :: SUB-DO-RULES
 UTILITY :: 1
 IF :: (R-QA = UK)
 THEN :: (RC = 99)

RULE199
 =====
 SUBJECT :: SUB-DO-RULES
 UTILITY :: 99
 IF :: ((PASS OR R-SURF = Ok OR R-DC = Ok) AND
 BLOWER IS KNOWN AND QAIR = 999)
 THEN :: (R-QA = UK)

RULE200
 =====
 SUBJECT :: SUB-DO-RULES
 UTILITY :: 0
 IF :: (R-QA = N/A)
 THEN :: (RC = 100)

RULE201
 =====
 SUBJECT :: SUB-DO-RULES
 UTILITY :: 100
 IF :: ((PASS OR R-SURF = Ok OR R-DC = Ok) AND
 SEARCH = Limited) THEN :: (R-QA = N/A)

RULE202
 =====
 SUBJECT :: SUB-DO-RULES
 UTILITY :: -1
 IF :: (R-DC = Ok)
 THEN :: (RC = 101 CF 60)

RULE203
 =====
 SUBJECT :: SUB-DO-RULES
 UTILITY :: 97
 IF :: ((PASS OR R-SURF = Not Uni) AND DIFFUSER <=
 DC-O) THEN :: (R-DC = Ok)

```

RULE204
=====
SUBJECT :: SUB-DO-RULES
UTILITY :: -2
IF :: (R-DC = Dirty)
THEN :: (RC = 102 CF 70)

RULE205
=====
SUBJECT :: SUB-DO-RULES
UTILITY :: 98
IF :: ((PASS OR R-SURF = Not Uni) AND DIFFUSER >
DC-O) THEN :: (R-DC = Dirty)

RULE206
=====
SUBJECT :: SUB-DO-RULES
UTILITY :: -3
IF :: (R-DC = UK)
THEN :: (RC = 103)

RULE207
=====
SUBJECT :: SUB-DO-RULES
UTILITY :: 99
IF :: ((PASS OR R-SURF = Not Uni) AND DIFFUSER =
999) THEN :: (R-DC = UK)

RULE208
=====
SUBJECT :: SUB-DO-RULES
UTILITY :: -4
IF :: (R-DC = N/A)
THEN :: (RC = 104)

RULE209
=====
SUBJECT :: SUB-DO-RULES
UTILITY :: 100
IF :: ((PASS OR R-SURF = Not Uni) AND SEARCH =
Limited) THEN :: (R-DC = N/A)

RULE210
=====
SUBJECT :: SUB-DO-RULES
UTILITY :: -5
IF :: (R-SURF = Ok)
THEN :: (RC = 105 CF 50)

RULE211
=====
SUBJECT :: SUB-DO-RULES
IF :: ((PASS OR R-DIV5 = No OR R-FM2 = Low OR R-FM2
= Norm) AND R-SURF IS NOTKNOWN AND SURFACE =
"Uniform") THEN :: (R-SURF = Ok)

RULE212
=====
SUBJECT :: SUB-DO-RULES
UTILITY :: -6
IF :: (R-SURF = N/A)
THEN :: (RC = 106)

RULE213
=====
SUBJECT :: SUB-DO-RULES
IF :: ((PASS OR R-DIV5 = No OR R-FM2 = Low OR R-FM2
= Norm) AND SEARCH = "Limited")
THEN :: (R-SURF = N/A)

RULE214
=====
SUBJECT :: SUB-DO-RULES
UTILITY :: -7
IF :: (R-SURF = Not Uni)
THEN :: (RC = 107)

RULE215
=====
SUBJECT :: SUB-DO-RULES
IF :: ((PASS OR R-DIV5 = No OR R-FM2 = Low OR R-FM2
= Norm) AND R-SURF IS NOTKNOWN AND SURFACE =
"Not Uniform") THEN :: (R-SURF = Not Uni)

RULE216
=====
SUBJECT :: SUB-DO-RULES
UTILITY :: -8
IF :: (R-DIV5 = No)
THEN :: (RC = 108)

RULE217
=====
SUBJECT :: SUB-DO-RULES
UTILITY :: 82
IF :: (R-X3 = Ok AND R-DIV5 IS NOTKNOWN AND HOLD
= No) THEN :: (R-DIV5 = No)

RULE218
=====
SUBJECT :: SUB-DO-RULES
UTILITY :: -9
IF :: (R-DIV5 = Yes)
THEN :: (RC = 109)

RULE219
=====
SUBJECT :: SUB-DO-RULES
UTILITY :: 83
IF :: (R-X3 = Ok AND R-DIV5 IS NOTKNOWN AND HOLD
= Yes) THEN :: (R-DIV5 = Yes)

RULE220
=====
SUBJECT :: SUB-DO-RULES
UTILITY :: -10
IF :: (R-DIV5 = UK)
THEN :: (RC = 110)

RULE221
=====
SUBJECT :: SUB-DO-RULES
UTILITY :: 84
IF :: (R-X3 = Ok AND R-DIV5 IS NOTKNOWN AND HOLD
= Not Sure) THEN :: (R-DIV5 = UK)

RULE222
=====
SUBJECT :: SUB-DO-RULES
UTILITY :: -11
IF :: (R-DIV5 = N/A)
THEN :: (RC = 111)

RULE223
=====
SUBJECT :: SUB-DO-RULES
UTILITY :: 85
IF :: (R-X3 = Ok AND SEARCH = Limited)
THEN :: (R-DIV5 = N/A)

```



```

RULE224
=====
SUBJECT :: SUB-DO-RULES
UTILITY :: -12
IF :: (R-X3 = Low)
THEN :: (RC = 112 CF 95)

RULE225
=====
SUBJECT :: SUB-DO-RULES
IF :: ((PASS OR R-FM2 = High) AND X < MLSS-S)
THEN :: (R-X3 = Low CF (CERTAINTY X))

RULE226
=====
SUBJECT :: SUB-DO-RULES
UTILITY :: -13
IF :: (R-X3 = Ok)
THEN :: (RC = 113 CF 90)

RULE227
=====
SUBJECT :: SUB-DO-RULES
IF :: ((PASS OR R-FM2 = High) AND X >= MLSS-S)
THEN :: (R-X3 = Ok CF (CERTAINTY X))

RULE228
=====
SUBJECT :: SUB-DO-RULES
UTILITY :: -14
IF :: (R-X3 = UK)
THEN :: (RC = 114)

RULE229
=====
SUBJECT :: SUB-DO-RULES
IF :: ((PASS OR R-FM2 = High) AND X = UNKNOWN)
THEN :: (R-X3 = UK)

RULE230
=====
SUBJECT :: SUB-DO-RULES
UTILITY :: -15
IF :: (R-FM2 = Low)
THEN :: (RC = 115 CF 90)

RULE231
=====
SUBJECT :: SUB-DO-RULES
IF :: ((PASS OR R-MBA = No) AND FM < FM-L)
THEN :: (R-FM2 = Low CF (CERTAINTY FM))

RULE232
=====
SUBJECT :: SUB-DO-RULES
UTILITY :: -16
IF :: (R-FM2 = High)
THEN :: (RC = 116 CF 90)

RULE233
=====
SUBJECT :: SUB-DO-RULES
IF :: ((PASS OR R-MBA = No) AND FM > FM-N)
THEN :: (R-FM2 = High CF (CERTAINTY FM))

RULE234
=====
SUBJECT :: SUB-DO-RULES
UTILITY :: -17
IF :: (R-FM2 = UK)
THEN :: (RC = 117)

RULE235
=====
SUBJECT :: SUB-DO-RULES
IF :: ((PASS OR R-MBA = No) AND FM = UNKNOWN)
THEN :: (R-FM2 = UK)

RULE236
=====
SUBJECT :: SUB-DO-RULES
UTILITY :: -18
IF :: (R-FM2 = Norm)
THEN :: (RC = 118 CF 95)

RULE237
=====
SUBJECT :: SUB-DO-RULES
IF :: ((PASS OR R-MBA = No) AND FM >= FM-L AND
FM <= FM-N) THEN :: (R-FM2 = Norm CF (CERTAINTY
FM))

RULE238
=====
SUBJECT :: SUB-DO-RULES
UTILITY :: -19
IF :: (R-MBA = Yes)
THEN :: (RC = 119)

RULE239
=====
SUBJECT :: SUB-DO-RULES
IF :: ((R-DO = Low OR INCREASE-DO) AND R-MBA IS
NOTKNOWN AND MBA = " Yes" )
THEN :: (R-MBA = Yes)

RULE240
=====
SUBJECT :: SUB-DO-RULES
UTILITY :: -20
IF :: (R-MBA = UK)
THEN :: (RC = 120)

RULE241
=====
SUBJECT :: SUB-DO-RULES
IF :: ((R-DO = Low OR INCREASE-DO) AND R-MBA IS
NOTKNOWN AND MBA = " Not Sure" )
THEN :: (R-MBA = UK)

RULE242
=====
SUBJECT :: SUB-DO-RULES
UTILITY :: -21
IF :: (R-MBA = N/A)
THEN :: (RC = 121)

RULE243
=====
SUBJECT :: SUB-DO-RULES
IF :: ((R-DO = Low OR INCREASE-DO) AND SEARCH =
Limited) THEN :: (R-MBA = N/A)

RULE244
=====
SUBJECT :: SUB-DO-RULES
UTILITY :: -22
IF :: (R-MBA = No)
THEN :: (RC = 122)

```

```

RULE245
=====
SUBJECT :: SUB-DO-RULES
IF :: ((R-DO = Low OR INCREASE-DO) AND R-MBA IS
NOTKNOWN AND MBA = "No" )
THEN :: (R-MBA = No)

```

```

=====
Frame :: PH
=====

```

```

IDENTIFIER :: "PH-"
PARENTS :: (NITRIFICATION ASPX)
GOALS :: (R-PH)
PARMGROUP :: PH-PARMS
RULEGROUPS :: (PH-RULES)
OFFSPRING :: (SUB-PH)
PH-PARMS :: (R-PH)
PH-RULES :: (RULE272 RULE273 RULE274 RULE275
RULE276 RULE277 RULE278 RULE279 )

```

```

=====
PH-RULES
=====

```

```

RULE272
=====
SUBJECT :: PH-RULES
UTILITY :: -37
IF :: (R-PH = High)
THEN :: (RC = 137 CF 95)

```

```

RULE273
=====
SUBJECT :: PH-RULES
IF :: (CURPH > = PH-H)
THEN :: (R-PH = High CF (CERTAINTY CURPH))

```

```

RULE274
=====
SUBJECT :: PH-RULES
UTILITY :: -38
IF :: (R-PH = Low)
THEN :: (RC = 138 CF 95)

```

```

RULE275
=====
SUBJECT :: PH-RULES
IF :: (CURPH < = PH-L)
THEN :: (R-PH = Low CF (CERTAINTY CURPH))

```

```

RULE276
=====
SUBJECT :: PH-RULES
UTILITY :: -39
IF :: (R-PH = UK)
THEN :: (RC = 139)

```

```

RULE277
=====
SUBJECT :: PH-RULES
IF :: (CURPH = UNKNOWN)
THEN :: (R-PH = UK)

```

```

RULE278
=====
SUBJECT :: PH-RULES
UTILITY :: -40
IF :: (R-PH = Ok)
THEN :: (RC = 140 CF 90)

```

```

RULE279
=====
SUBJECT :: PH-RULES
IF :: (CURPH > PH-L AND CURPH < PH-H)
THEN :: (R-PH = Ok CF (CERTAINTY CURDO))

```

```

=====
Frame :: SUB-PH
=====

```

```

IDENTIFIER :: "SUB-PH-"
PARENTS :: (PH NITRIFICATION ASPX)
PROMPT1ST :: PREMISE
PREMISE :: ($AND
($OR
(SAME FRAME PASS)
(SAME FRAME R-PH "Low")
(SAME FRAME R-PH "High")))
PARMGROUP :: SUB-PH-PARMS
RULEGROUPS :: (SUB-PH-RULES)
SUB-PH-PARMS :: (CAUSTIC R-CA R-IPH)
SUB-PH-RULES :: (RULE254 RULE255 RULE256 RULE257
RULE258 RULE259 RULE260 RULE261 RULE262 RULE263
RULE264 RULE265 RULE266 RULE267 RULE268 RULE269
RULE270 RULE271 )

```

```

=====
SUB-PH-RULES
=====

```

```

RULE254
=====
SUBJECT :: SUB-PH-RULES
UTILITY :: -28
IF :: (R-IPH = Low)
THEN :: (RC = 128 CF 95)

```

```

RULE255
=====
SUBJECT :: SUB-PH-RULES
IF :: ((PASS OR R-CA = Ok OR R-CA = N/A) AND IPH
< = IPH-L) THEN :: (R-IPH = Low CF (CERTAINTY IPH))

```

```

RULE256
=====
SUBJECT :: SUB-PH-RULES
UTILITY :: -29
IF :: (R-IPH = Ok)
THEN :: (RC = 129 CF 90)

```

```

RULE257
=====
SUBJECT :: SUB-PH-RULES
IF :: ((PASS OR R-CA = Ok OR R-CA = N/A) AND IPH >
IPH-L AND IPH < IPH-H )
THEN :: (R-IPH = Ok CF (CERTAINTY IPH))

```

```

RULE258
=====
SUBJECT :: SUB-PH-RULES
UTILITY :: -30
IF :: (R-IPH = High)
THEN :: (RC = 130 CF 95)

RULE259
=====
SUBJECT :: SUB-PH-RULES
IF :: ((PASS OR R-CA = Ok OR R-CA = N/A) AND IPH
> = IPH-H) THEN :: (R-IPH = High CF (CERTAINTY IPH))

RULE260
=====
SUBJECT :: SUB-PH-RULES
UTILITY :: -31
IF :: (R-IPH = UK)
THEN :: (RC = 131)

RULE261
=====
SUBJECT :: SUB-PH-RULES
IF :: ((PASS OR R-CA = Ok OR R-CA = N/A) AND IPH =
UNKNOWN) THEN :: (R-IPH = UK)

RULE262
=====
SUBJECT :: SUB-PH-RULES
UTILITY :: -32
IF :: (R-CA = High)
THEN :: (RC = 132 CF 95)

RULE263
=====
SUBJECT :: SUB-PH-RULES
IF :: ((PASS OR R-PH = High) AND R-CA IS NOTKNOWN
AND CAUSTIC > CALC-PH)
THEN :: (R-CA = High)

RULE264
=====
SUBJECT :: SUB-PH-RULES
UTILITY :: -33
IF :: (R-CA = Low)
THEN :: (RC = 133 CF 98)

RULE265
=====
SUBJECT :: SUB-PH-RULES
IF :: ((PASS OR R-PH = Low) AND R-CA IS NOTKNOWN
AND CAUSTIC < CALC-PH)
THEN :: (R-CA = Low)

RULE266
=====
SUBJECT :: SUB-PH-RULES
UTILITY :: -34
IF :: (R-CA = Ok)
THEN :: (RC = 134 CF 95)

RULE267
=====
SUBJECT :: SUB-PH-RULES
IF :: (((PASS OR R-PH = High) AND R-CA IS NOTKNOWN
AND CAUSTIC <= CALC-PH) OR ((PASS OR R-PH = Low)
AND R-CA IS NOTKNOWN AND CAUSTIC >= CALC-PH)
)
THEN :: (R-CA = Ok)

```

```

RULE268
=====
SUBJECT :: SUB-PH-RULES
UTILITY :: -35
IF :: (R-CA = UK)
THEN :: (RC = 135)

RULE269
=====
SUBJECT :: SUB-PH-RULES
UTILITY :: 99
IF :: ((PASS OR R-PH = Low OR R-PH = High) AND R-CA
IS NOTKNOWN AND CAUSTIC = 999 )
THEN :: (R-CA = UK)

RULE270
=====
SUBJECT :: SUB-PH-RULES
UTILITY :: -36
IF :: (R-CA = N/A)
THEN :: (RC = 136)

RULE271
=====
SUBJECT :: SUB-PH-RULES
UTILITY :: 100
IF :: ((PASS OR R-PH = Low OR R-PH = High) AND
SEARCH = Limited) THEN :: (R-CA = N/A)

=====
Frame :: PO4
=====

IDENTIFIER :: "PO4-"
PARENTS :: (NITRIFICATION ASPX)
GOALS :: (R-PO4)
PARMGROUP :: PO4-PARMS
RULEGROUPS :: (PO4-RULES)
OFFSPRING :: (SUB-PO4)
PO4-PARMS :: (R-PO4)
PO4-RULES :: (RULE300 RULE301 RULE302 RULE303
RULE304 RULE305)

=====
PO4-RULES
=====
RULE300
=====
SUBJECT :: PO4-RULES
UTILITY :: -51
IF :: (R-PO4 = Low)
THEN :: (RC = 151 CF 95)
RULE301
=====
SUBJECT :: PO4-RULES
IF :: (PO4OUT < PO4-L)
THEN :: (R-PO4 = Low CF (CERTAINTY PO4OUT))
RULE302
=====
SUBJECT :: PO4-RULES
UTILITY :: -52
IF :: (R-PO4 = UK)
THEN :: (RC = 152)

```

RULE303
 =====
 SUBJECT :: PO4-RULES
 IF :: (PO4OUT = UNKNOWN)
 THEN :: (R-PO4 = UK)

RULE304
 =====
 SUBJECT :: PO4-RULES
 UTILITY :: -53
 IF :: (R-PO4 = Ok)
 THEN :: (RC = 153 CF 95)

RULE305
 =====
 SUBJECT :: PO4-RULES
 IF :: (PO4OUT >= PO4-L)
 THEN :: (R-PO4 = Ok CF (CERTAINTY PO4OUT))

=====
 Frame :: SUB-PO4
 =====

IDENTIFIER :: "SUB-PO4-"
 PARENTS :: (PO4 NITRIFICATION ASPX)
 PROMPT1ST :: PREMISE
 PREMISE :: (\$AND
 (\$OR
 (SAME FRAME PASS)
 (SAME FRAME R-PO4 "Low")))
 PARGROUP :: SUB-PO4-PARMS
 RULEGROUPS :: (SUB-PO4-RULES)
 SUB-PO4-PARMS :: (PADD R-EPO4 R-EPO42 R-PADD)
 SUB-PO4-RULES :: (RULE280 RULE281 RULE282 RULE283
 RULE284 RULE285 RULE286 RULE287 RULE288 RULE289
 RULE290 RULE291 RULE292 RULE293 RULE294 RULE295
 RULE296 RULE297 RULE298 RULE299)

=====
 SUB-PO4-RULES
 =====

RULE280
 =====
 SUBJECT :: SUB-PO4-RULES
 UTILITY :: -41
 IF :: (R-EPO4 = Low)
 THEN :: (RC = 141 CF 98)

RULE281
 =====
 SUBJECT :: SUB-PO4-RULES
 IF :: (((PASS AND R-PADD != Ok) OR R-PADD = Low)
 AND PADD < EPO4) THEN :: (R-EPO4 = Low CF
 (CERTAINTY EPO4))

RULE282
 =====
 SUBJECT :: SUB-PO4-RULES
 UTILITY :: -42
 IF :: (R-EPO4 = Ok)
 THEN :: (RC = 142 CF 90)

RULE283
 =====
 SUBJECT :: SUB-PO4-RULES
 IF :: (((PASS AND R-PADD != Ok) OR R-PADD = Low)
 AND PADD >= EPO4) THEN :: (R-EPO4 = Ok CF
 (CERTAINTY EPO4))

RULE284
 =====
 SUBJECT :: SUB-PO4-RULES
 UTILITY :: -43
 IF :: (R-EPO4 = UK)
 THEN :: (RC = 143)

RULE285
 =====
 SUBJECT :: SUB-PO4-RULES
 IF :: (((PASS AND R-PADD != Ok) OR R-PADD = Low)
 AND EPO4 = UNKNOWN) THEN :: (R-EPO4 = UK)

RULE286
 =====
 SUBJECT :: SUB-PO4-RULES
 UTILITY :: -44
 IF :: (R-EPO42 = Low)
 THEN :: (RC = 144 CF 98)

RULE287
 =====
 SUBJECT :: SUB-PO4-RULES
 IF :: (R-PADD = Ok AND PADD < EPO4)
 THEN :: (R-EPO42 = Low CF (CERTAINTY EPO4))

RULE288
 =====
 SUBJECT :: SUB-PO4-RULES
 UTILITY :: -45
 IF :: (R-EPO42 = Ok)
 THEN :: (RC = 145 CF 90)

RULE289
 =====
 SUBJECT :: SUB-PO4-RULES
 IF :: (R-PADD = Ok AND PADD >= EPO4)
 THEN :: (R-EPO42 = Ok CF (CERTAINTY EPO4))

RULE290
 =====
 SUBJECT :: SUB-PO4-RULES
 UTILITY :: -46
 IF :: (R-EPO42 = UK)
 THEN :: (RC = 146)

RULE291
 =====
 SUBJECT :: SUB-PO4-RULES
 IF :: (R-PADD = Ok AND EPO4 = UNKNOWN)
 THEN :: (R-EPO42 = UK)

RULE292
 =====
 SUBJECT :: SUB-PO4-RULES
 UTILITY :: -47
 IF :: (R-PADD = Low)
 THEN :: (RC = 147 CF 90)

RULE293
 =====
 SUBJECT :: SUB-PO4-RULES
 IF :: ((PASS OR R-PO4 = Low) AND R-PADD IS
 NOTKNOWN AND PADD < PADD-L)
 THEN :: (R-PADD = Low)

```

RULE294
=====
SUBJECT :: SUB-PO4-RULES
UTILITY :: -48
IF :: (R-PADD = N/A)
THEN :: (RC = 148)

RULE295
=====
SUBJECT :: SUB-PO4-RULES
UTILITY :: 100
IF :: ((PASS OR R-PO4 = Low) AND SEARCH = Limited)
THEN :: (R-PADD = N/A)

RULE296
=====
SUBJECT :: SUB-PO4-RULES
UTILITY :: -49
IF :: (R-PADD = UK)
THEN :: (RC = 149)

RULE297
=====
SUBJECT :: SUB-PO4-RULES
UTILITY :: 99
IF :: ((PASS OR R-PO4 = Low) AND R-PADD IS
NOTKNOWN AND PADD = 999) THEN :: (R-PADD = UK)

RULE298
=====
SUBJECT :: SUB-PO4-RULES
UTILITY :: -50
IF :: (R-PADD = Ok)
THEN :: (RC = 150 CF 90)

RULE299
=====
SUBJECT :: SUB-PO4-RULES
IF :: ((PASS OR R-PO4 = Low) AND R-PADD IS
NOTKNOWN AND PADD >= PADD-L)
THEN :: (R-PADD = Ok)

```

```

RULE323
=====
SUBJECT :: TEMPERATURE-RULES
IF :: (TEMP >= TEMP-H)
THEN :: (R-TEMP = High CF (CERTAINTY TEMP))

```

```

RULE324
=====
SUBJECT :: TEMPERATURE-RULES
UTILITY :: -63
IF :: (R-TEMP = Low)
THEN :: (RC = 163 CF 60)

```

```

RULE325
=====
SUBJECT :: TEMPERATURE-RULES
IF :: (TEMP <= TEMP-L)
THEN :: (R-TEMP = Low CF (CERTAINTY TEMP))

```

```

RULE326
=====
SUBJECT :: TEMPERATURE-RULES
UTILITY :: -64
IF :: (R-TEMP = UK)
THEN :: (RC = 164)

```

```

RULE327
=====
SUBJECT :: TEMPERATURE-RULES
IF :: (TEMP = UNKNOWN)
THEN :: (R-TEMP = UK)

```

```

RULE328
=====
SUBJECT :: TEMPERATURE-RULES
UTILITY :: -65
IF :: (R-TEMP = Ok)
THEN :: (RC = 165 CF 60)

```

```

RULE329
=====
SUBJECT :: TEMPERATURE-RULES
IF :: (TEMP > TEMP-L AND TEMP < TEMP-H)
THEN :: (R-TEMP = Ok CF (CERTAINTY TEMP))

```

```

=====
Frame :: TEMPERATURE
=====

```

```

IDENTIFIER :: "TEMPERATURE-"
PARENTS :: (NITRIFICATION ASPX)
GOALS :: (R-TEMP)
PARMGROUP :: TEMPERATURE-PARMS
RULEGROUPS :: (TEMPERATURE-RULES)
OFFSPRING :: (SUB-TEMP)
TEMPERATURE-PARMS :: (R-TEMP)
TEMPERATURE-RULES :: (RULE322 RULE323 RULE324
RULE325 RULE326 RULE327 RULE328 RULE329)

```

```

=====
TEMPERATURE-RULES
=====
RULE322
=====
SUBJECT :: TEMPERATURE-RULES
UTILITY :: -62
IF :: (R-TEMP = High)
THEN :: (RC = 162 CF 90)

```

```

=====
Frame :: SUB-TEMP
=====

```

```

IDENTIFIER :: "SUB-TEMP-"
PARENTS :: (TEMPERATURE NITRIFICATION ASPX)
PROMPT1ST :: PREMISE
PREMISE :: ($AND
($OR
(SAME FRAME PASS)
(SAME FRAME R-TEMP "Low")))
PARMGROUP :: SUB-TEMP-PARMS
RULEGROUPS :: (SUB-TEMP-RULES)
SUB-TEMP-PARMS :: (R-ITEMP)
SUB-TEMP-RULES :: (RULE306 RULE307 RULE308 RULE309
RULE310 RULE311 RULE312 RULE313 RULE314 RULE315
RULE316 RULE317 RULE318 RULE319 RULE320 RULE321)

```

```

=====
SUB-TEMP-RULES
=====

RULE306
=====
SUBJECT :: SUB-TEMP-RULES
UTILITY :: -54
IF :: (R-DIV6 = Yes)
THEN :: (RC = 154)

RULE307
=====
SUBJECT :: SUB-TEMP-RULES
UTILITY :: 78
IF :: ((PASS OR R-ITEMP = Low OR R-ITEMP = High)
AND R-DIV6 IS NOTKNOWN AND HOLD = "Yes" )
THEN :: (R-DIV6 = Yes)

RULE308
=====
SUBJECT :: SUB-TEMP-RULES
UTILITY :: -55
IF :: (R-DIV6 = No)
THEN :: (RC = 155)

RULE309
=====
SUBJECT :: SUB-TEMP-RULES
UTILITY :: 79
IF :: ((PASS OR R-ITEMP = Low OR R-ITEMP = High)
AND R-DIV6 IS NOTKNOWN AND HOLD = "No" )
THEN :: (R-DIV6 = No)

RULE310
=====
SUBJECT :: SUB-TEMP-RULES
UTILITY :: -56
IF :: (R-DIV6 = UK)
THEN :: (RC = 156)

RULE311
=====
SUBJECT :: SUB-TEMP-RULES
UTILITY :: 80
IF :: ((PASS OR R-ITEMP = Low OR R-ITEMP = High)
AND R-DIV6 IS NOTKNOWN AND HOLD = "Not Sure" )
THEN :: (R-DIV6 = UK)

RULE312
=====
SUBJECT :: SUB-TEMP-RULES
UTILITY :: -57
IF :: (R-DIV6 = N/A)
THEN :: (RC = 157)

RULE313
=====
SUBJECT :: SUB-TEMP-RULES
UTILITY :: 81
IF :: ((PASS OR R-ITEMP = Low OR R-ITEMP = High)
AND SEARCH = "Limited" )
THEN :: (R-DIV6 = N/A)

RULE314
=====
SUBJECT :: SUB-TEMP-RULES
UTILITY :: -58
IF :: (R-ITEMP = UK)
THEN :: (RC = 158)

RULE315
=====
SUBJECT :: SUB-TEMP-RULES
IF :: ((PASS OR R-TEMP = High OR R-TEMP = Low) AND
ITEMP = UNKNOWN) THEN :: (R-ITEMP = UK)

RULE316
=====
SUBJECT :: SUB-TEMP-RULES
UTILITY :: -59
IF :: (R-ITEMP = ?)
THEN :: (RC = 159 CF 75)

RULE317
=====
SUBJECT :: SUB-TEMP-RULES
IF :: (((PASS OR R-TEMP = Low) AND ITEMTEMP >=
ITEMTEMP-H) OR ((PASS OR R-TEMP = "High" ) AND ITEMTEMP
<= ITEMTEMP-L ) ) THEN :: (R-ITEMP = ? CF (CERTAINTY
ITEMP))

RULE318
=====
SUBJECT :: SUB-TEMP-RULES
UTILITY :: -60
IF :: (R-ITEMP = Low)
THEN :: (RC = 160 CF 80)

RULE319
=====
SUBJECT :: SUB-TEMP-RULES
IF :: ((PASS OR R-TEMP = Low) AND ITEMTEMP < ITEMTEMP-H)
THEN :: (R-ITEMP = Low CF (CERTAINTY ITEMTEMP))

RULE320
=====
SUBJECT :: SUB-TEMP-RULES
UTILITY :: -61
IF :: (R-ITEMP = High)
THEN :: (RC = 161 CF 90)

RULE321
=====
SUBJECT :: SUB-TEMP-RULES
IF :: ((PASS OR R-TEMP = High) AND ITEMTEMP > ITEMTEMP-L)
THEN :: (R-ITEMP = High CF (CERTAINTY ITEMTEMP))

=====
Frame :: SETTLING-PROBLEMS
=====

IDENTIFIER :: SETTLING-PROBLEMS-
PARENTS :: (ASFX)
PROMPTIST :: PREMISE
PREMISE :: ($AND
(SAME FRAME CHECK-SSP))
PARMGROUP :: SETTLING-PROBLEMS-PARMS
RULEGROUPS :: (SETTLING-PROBLEMS-RULES)
SETTLING-PROBLEMS-PARMS :: (FOAM)
SETTLING-PROBLEMS-RULES :: (RULE330 RULE331
RULE332 RULE333 RULE334 RULE335 RULE336 RULE337)

```

=====
SETTLING-PROBLEMS-RULES
=====

RULE330
=====
SUBJECT :: SETTLING-PROBLEMS-RULES
IF :: (MCRT > SA-D OR CURDO > 4)
THEN :: (SSP = 166 CF 60)

RULE331
=====
SUBJECT :: SETTLING-PROBLEMS-RULES
IF :: (FM < FM-L OR IR > IR-N OR QW < OPTQW)
THEN :: (SSP = 167 CF 60)

RULE332
=====
SUBJECT :: SETTLING-PROBLEMS-RULES
IF :: (MCRT < SA-D OR FM > FM-N)
THEN :: (SSP = 168 CF 60)

RULE333
=====
SUBJECT :: SETTLING-PROBLEMS-RULES
IF :: (QW < OPTQW AND SBD > SBD-H)
THEN :: (SSP = 169 CF 60)

RULE334
=====
SUBJECT :: SETTLING-PROBLEMS-RULES
IF :: ((MCRT < SA-D OR X < MLSS-L OR FM > FM-N)
AND SEARCH != "Limited" AND FOAM = "White Foam")
THEN :: (SSP = 170 CF 70)

RULE335
=====
SUBJECT :: SETTLING-PROBLEMS-RULES
IF :: (FM < FM-L AND SEARCH != Limited AND FOAM
= Thick Brown Foam) THEN :: (SSP = 171 CF 70)

RULE336
=====
SUBJECT :: SETTLING-PROBLEMS-RULES
IF :: (CURDO < DO-L AND SEARCH != Limited AND
FOAM = Black Foam) THEN :: (SSP = 172 CF 70)

RULE337
=====
SUBJECT :: SETTLING-PROBLEMS-RULES
IF :: (SEARCH != "Limited" AND (FOAM = "Light Brown
Foam" OR FOAM = " Not Sure"))
THEN :: (SSP = 173 CF 70)

UNIVERSITY OF CALIFORNIA
Los Angeles

Development of a
Knowledge-Based Expert System for
Process Control of Nitrification in the
Activated Sludge Process

SUPPLEMENT
ASPX Program Code

by

Naci H. Ozgur

1991

Table of Contents

ASPX Program Code	1
Effluent ammonia simulation program code (Fourth Order Runge-Kutta MCRT Simulation)	58
Nitrification state determination program code	63

ASPX PROGRAM CODE FOR TURBO PASCAL 5.5

(\$M 1800, 0, 172900)

Uses Crt, Graph, Dos, GrDrivers, GrFonts, GrMenus,
Basics, IO, Plot, MenuEXT, States, NameTool;

Label 1,2,3,4;

Var Kare: Boolean;

BEGIN

Altered:= False;
Erase:= False;
LimitsAltered:= False;
StatesAltered:= False;
NewMax:= False;
Kare:= False;
Error:= False;

Init;

2: Fill.Color:= Blue;
ClearViewPort;
SetFillStyle(SolidFill,Blue);
Bar(0,0,GetMaxX,GetMaxY);
Beginning;
MainScreen;
ValidChoices:= [1..8];
If Secenek IN [20,21] Then begin; PrimaryMenu(1); ShowCursor;

Goto 1; end;

If ParamCount= 0 Then UserNames
Else begin; PrimaryMenu(1);
Assign(dt,'CURRENT.USR'); Reset(dt);
Readln(dt,User);
Close(dt);
SecondaryMenu(4,'Revise Conclusion

Statements');

end;

UserAccess:= 'Unlimited';
ShowCursor;

1: If Altered Then UpdateCalcs;
If StatesAltered Then UpdateFactors;
If NewMax Then Ciz;

ReadMouse;
OldSecenek:= Secenek;
Secenek:= 0;
If (b = 1) Then FindSecenek;
If (b = 2) Then Goto 1;

4.;

Case Secenek Of

0: If Kare Then begin; DrawSquareX; Delay(5); Goto 1; end
Else Goto 1;

```

1: PrimaryMenu(1);
2: PrimaryMenu(2);
3: PrimaryMenu(3);
4: PrimaryMenu(99);
5: OutTextxy(0,250,'                HELP ME!!!');
6: { Glossary };
7: begin; PrimaryMenu(2); Secenek:= 25; Goto 4; end;
8: begin; PrimaryMenu(1); Secenek:= 19; Goto 4; end;
9..14: { Open };
15..18: {Report Generation};
19: begin; Kare:= true; SecondaryMenu(1,'Select Train & Run');
end;
20: begin; SwapVectors;
    Exec('C:\Command.Com','/C STATE.EXE');
    SwapVectors;
    Readln;
    Goto 2; end;
21: begin; SwapVectors;
    Exec('C:\Command.Com','/C NH4-SIM.EXE');
    SwapVectors;
    Goto 2; end;
22: SecondaryMenu(2,'Set Limits');
23: SecondaryMenu(3,'Set Nitrification States');
24: SecondaryMenu(4,'Revise Conclusion Statements');
25: SecondaryMenu(5,'Data base records');
26: begin; Kare:= True; SecondaryMenu(6,'Pattern Analysis'); end;
27: begin; Kare:= True; SecondaryMenu(8,'DB Statistics'); end;
28: begin; Kare:= True; SecondaryMenu(9,'Transfer DB files');
end;
29: begin; Kare:= True; SecondaryMenu(10,'Confirm DB records');
end;
30..32: Goto 3;
33: Confirmation;
34: begin; ShowTrain:= 35; UpdateCalcs; end; {Select ASP700}
35: begin; ShowTrain:= 34; UpdateCalcs; end; {Select ASP600}
36: begin; MainTrain; PrintMenuItems; end;
37: ASPTrain;
38..122: begin; Altered:= True; ReadNumber; end;
123..149: begin; LimitsAltered:= True; ReadNumber; end;
150..160: { Open };
161..183: If Erase Then GetNametoErase
        Else PasswordCheck('U');
184, 185: DontDeleteName;
186: DeleteName;
187: begin; NewMax:= True; StatesAltered:= True; ReadNumber; end;
188: MoveSTDLine;
189: MoveLine(1);
190: Moveline(2);
191: Moveline(3);
192..199: begin; StatesAltered:= True; ReadNumber; end;
200..201: PasswordCheck('S');
213: begin; Secenek:= OldSecenek; Goto 4; end;
214: Goto 3;
215..219: { Tools for Review };

```

```

220,223: Increase;
221,224: Decrease;
222,225: begin; CleanArrows; UpdateCalcs; end;
226..228: Change('Date');
229: Change('Time');
230: begin; Train:= 230; XX('E',41); XX('D',42); end;
231: begin; Train:= 231; XX('E',42); XX('D',41); end;
232: begin; SearchTech:= 232; XX('E',44); XX('E',45); XX('D',43);
end;
233: begin; SearchTech:= 233; XX('E',43); XX('E',45); XX('D',44);
end;
234: begin; SearchTech:= 234; XX('E',43); XX('E',44); XX('D',45);
end;
235..246: { Open };
247: ReadNumber;
248: SecondaryMenu(7,'Draw Plot');
249..254: begin; CleanScreen;
      If Kare Then Kare:= False;
      If Error Then Error:= False; end;
255: begin; If Altered      Then WriteDataBase;
      If LimitsAltered Then WriteLimits;
      If StatesAltered Then WriteStates;
      Kare:= False;
      If MenuOut= 22 Then begin; MenuOut:= 2; Secenek:= 26;
Goto 4; end;
      CleanScreen; end;
      end;
      Goto 1;

3: WriteParameterValues;
RestoreCrtMode;
Case Secenek Of
  30: begin; CloseName; Halt(2); end;           { Logout }
  31: begin; CloseName; Halt(1); end;           { Exit to DOS }
214: Halt(1);
  32: begin; WriteSearchTech; Halt(3); end;     { Run ES }
end;
END.

```

Unit GrMenus;

Interface

Uses Crt, Dos, Graph;

Type

ByteArray = Array [0..0] of Byte;

ByteArrayPtr= ^ByteArray;

AnyString = String[80];

RegPack = Record

Case Byte of

1 : (AX, BX, CX, DX, BP, SI, DI, DS, ES, Flags : integer);

```

                2 : (AL, AH, BL, BH, CL, CH, DL, DH : byte); end;
RegisterSet = Record
                AX, BX, CX, DX, BP, DI, SI, DS, ES, Flags: integer;
end;
    Location = Record
                XY: Array [1..4] of Integer;
                S: String[40];
                Fg, Bg: Byte; end;

DataBaseRecord = Record
                Date: String[8];
                Time: String[2];
                Values: Array [38..122] Of String[7]; end;

IndividualRecord = Record
                Date: String[8];
                Time: String[2];
                Values: Array [1..8] Of String[7]; end;

IndividualDBFile= File of IndividualRecord;

    Dortgen = Record
                XY: Array [1..4] of Integer; end;

Const
    MouseInt = $33;
    Days: Array [0..6] of String[3] = ('Sun', 'Mon', 'Tue', 'Wed', 'Thu',
    'Fri', 'Sat');

Var
                Regs: RegPack;
    Screen, Train, n, gd, gm: Integer;
ii, i, j, k, Command, Secenek: Integer;
    SearchTech, OldSecenek: Integer;
                OldChosen: Integer;
                mx, my, b: Integer;
    Std, Max, Tic, Tane: Integer;
    ShowTrain, Yeri, top: Integer;
                Color: Byte;
                DataBase: Array[1..5,1..80] of Integer;
                State, Property: Array[1..4] Of Integer;
    ValidChoices, OldChoices: Set Of 1..255;
    ValidSquares, OldSquares: Set Of 1..100;
                Required: Set Of 38..122;
                Choices: Array [0..255] of Location;
    RecordNumber, OldChoice: Integer;
                MenuOut, OldOut: Byte;
    Year, Month, Day, DayofWeek: Word;
    Hour, Minute, Second, Sec100: Word;
                c: Char;
    UserAccess, User, nfs, nf: String[20];
                Ay, Gun, Saat: String[2];
                Yil: String[4];
                Labelit: String[6];
                Suggestion: String[8];

```

```

        ConcText: AnyString;
        PassWord: Array [1..22] of String[15];
        TextStyle: TextSettingsType;
        Fill: FillSettingsType;
LimitsAltered, Altered, Ok: Boolean;
    Erase, StatesAltered: Boolean;
        NewMax: Boolean;
        dt: Text;
        Q, COD, XE, X: Real;
        QW, XW, QR, Null: Real;
        XSA, FM, OrgLoad, OptQw: Extended;
    OptRAS, OptRASP, CurRASP: Real;
        EBOD, EPO4: Real;
        PictureSize: Word;
        MaxPictureSize: LongInt;
        Image, Image2: ByteArrayPtr;
        Oldx, Oldy: integer;
        P: Array [1..3] of Integer;
Ucgen1, Ucgen2, Ucgen3, Ucgen4: Array [1..5] of PointType;
        DBRecord: Array [1..6] of DataBaseRecord;
        DBFile: File of DataBaseRecord;
        sonuc: array [1..30] of integer;
        curstate, find: string[80];
        Xed: Array [1..100] Of Boolean;
        Squares: Array [1..100] Of Dortgen;
DefLocation, DefDuration, DefType: Word;
        pH: Real;
        Error: boolean;
        Parameter: Array [1..5,1..93] of Real;
        IndRecord: IndividualRecord;
        IndDB: Array [1..5] of IndividualDBFile;
        FileName: String[80];

PROCEDURE ShowCursor;
PROCEDURE GetMouse(var X, Y, Button: integer);
PROCEDURE SetMouse(X, Y : integer);
PROCEDURE SetMinMaxXY(MinX, MaxX, MinY, MaxY : integer);
PROCEDURE SetMickeyRatio(Rx, Ry : integer);
PROCEDURE HideCursor;
PROCEDURE Init;
PROCEDURE GetKey;
PROCEDURE PrintXY(Pstring : AnyString; Fg, Bg: Byte; Col, Row :
Integer);
PROCEDURE BarF(x1,y1,x2,y2: Integer; Color:Byte);
PROCEDURE Clean(x1, y1, x2, y2: Integer);
PROCEDURE UseWin(x1, y1, x2, y2: Integer;
        Border, WinType, SplitY: Byte;
        Bcolor, Wincolor, SplitColor: Byte;
        Shadow: Boolean;
        Grow: Byte);
PROCEDURE MemorizeLocation(j, x, y: Integer; Color: Char);
PROCEDURE ReadMouse;

```

Implementation

```
PROCEDURE ShowCursor;
var Reg : RegisterSet;
begin
  Reg.AX := 1;
  Intr(MouseInt, Dos.Registers(Reg));
end;

PROCEDURE HideCursor;
var Reg : RegisterSet;
begin
  Reg.AX := 2;
  Intr(MouseInt, Dos.Registers(Reg));
end;

PROCEDURE GetMouse(var X, Y, Button: integer);
var Reg : RegisterSet;
begin
  with Reg do
  begin
    AX := 3;
    Intr(MouseInt, Dos.Registers(Reg));
    Button := BX;
    X := CX;
    Y := DX;
  end;
end;

PROCEDURE SetMouse(X, Y : integer);
var Reg : RegisterSet;
begin
  with Reg do
  begin
    AX := 4;
    CX := X;
    DX := Y;
    Intr(MouseInt, Dos.Registers(Reg));
  end;
end;

PROCEDURE SetMinMaxXY(MinX, MaxX, MinY, MaxY : integer);
var Reg : RegisterSet;
begin
  with Reg do
  begin
    AX := 7;
    CX := MinX;
    DX := MaxX;
    Intr(MouseInt, Dos.Registers(Reg));
    AX := 8;
    CX := MinY;
    DX := MaxY;
  end;
end;
```

```

    Intr(MouseInt, Dos.Registers(Reg));
end;
end;

PROCEDURE SetMickeyRatio(Rx, Ry : integer);
var Reg : RegisterSet;
begin
    with Reg do
        begin
            AX := 15;
            CX := Rx;
            DX := Ry;
            Intr(MouseInt, Dos.Registers(Reg));
        end;
    end;
end;

PROCEDURE Init;
begin; DetectGraph(gd, gm);
    If gd in [VGA, IBM8514] Then Else begin; Clrscr;
        Writeln('You need a VGA or higher graphics card');
Halt(1); end;
    If gd= IBM8514 Then gm:= IBM8514Lo;
        InitGraph(gd, gm, '');
        SetViewPort(0,0,GetmaxX,GetmaxY,True);
        SetMinMaxXY(0,650,0,GetMaxY-13);
        SetBkColor(Blue);
        SetTextStyle(0,0,0);
    end;

PROCEDURE GetKey;
begin; C := ReadKey;
    repeat
        if C = #0 then begin; C := ReadKey;
            if Ord(C) > 127 then C := #0
                else Command:= (Ord(C)-58);
        end
            else Command := Ord(C);
    until C <> #0;
end;

PROCEDURE BarF(x1,y1,x2,y2: Integer; Color:Byte);
Var DefColor: Byte;
begin; DefColor:= GetColor;
        SetFillStyle(SolidFill, Color);
        HideCursor;
        Bar(x1,y1,x2,y2);
        ShowCursor;
        SetColor(DefColor);
    end;

PROCEDURE PrintXY(Pstring : AnyString; Fg, Bg: Byte; Col, Row :
Integer);
begin; HideCursor;
        SetColor(Fg); SetFillStyle(Solidfill,Bg);

```



```

        GetTextSettings(TextStyle);
        If Row < 3 Then Row:= 600;
        If TextStyle.Font = 0 Then Bar(Col-2, Row-2,
Col+TextWidth(Pstring),
                                Row+TextHeight(Pstring)+2);
        If TextStyle.Font = 1 Then Bar(Col-2, Row+2,
Col+TextWidth(Pstring),
Row+Trunc(1.2*TextHeight(Pstring)));
        If TextStyle.Font = 2 Then Bar(Col-2, Row+2,
Col+TextWidth(Pstring),
Row+Trunc(1.2*TextHeight(Pstring)));
        OutTextxy(Col, Row, Pstring);
        ShowCursor;
    end;

PROCEDURE Clean(x1, y1, x2, y2: Integer);
begin; HideCursor;
        SetViewPort(x1,y1,x2,y2,true);
        ClearViewPort;
        SetViewPort(0,0,GetMaxX,GetMaxY,True);
        ShowCursor;
end;

PROCEDURE UseWin(x1, y1, x2, y2: Integer;
                Border, WinType, SplitY: Byte;
                Bcolor, Wincolor, SplitColor: Byte;
                Shadow: Boolean;
                Grow: Byte);
Var X,Y,IncX,IncY: real;
    i: Integer;
begin; Case Border Of
        1: SetLineStyle(SolidLn, 0, NormWidth);
        2: SetLineStyle(SolidLn, 0, ThickWidth);
        3: SetLineStyle(SolidLn, 0, ThickWidth);
    end;
    HideCursor;
    SetColor(Bcolor);
    Rectangle(x1, y1, x2, y2);
    SetFillStyle(SolidFill,Wincolor);
    X:= x2-(x2-x1)/2;   Y:= y2-(y2-y1)/2;
    IncX:= (x2-x1)/40; IncY:= (y2-y1)/40;
    Case Grow Of
        1: begin; For i:= 1 to 20 do begin;
Bar(Trunc(X-IncX*i)+2,Trunc(Y-IncY*i)+2,Trunc(X+IncX*i)-1,Trunc(Y+IncY*
i)-1);
                end; end;
        2: begin; For i:= 1 to 20 do begin;
Bar(Trunc(X-IncX*i),Trunc(Y-IncY),Trunc(X+IncX*i),Trunc(Y+IncY));
                Delay(5); end;
                For i:= 1 to 20 do begin;
Bar(Trunc(X-IncX*20),Trunc(Y-IncY*i),Trunc(X+IncX*20),Trunc(Y+IncY*i));
                end; end;
    end;
end;

```

```

end;

If Shadow Then begin;
  SetFillStyle(CloseDotFill, White);
  Bar(x1-10, y1+3, x1-2, y2+10);
  Bar(x1-2, y2+2, x2-3, y2+10); end;

If (WinType= 2) Then begin;
  SetFillStyle(SolidFill, SplitColor);
  If Border= 2 Then begin; Bar(x1+7, y1+7, x2-7, SplitY-2);
    Rectangle(x1+3, y1+3, x2-3,
y2-3);
    Rectangle(x1+7, y1+7, x2-7,
SplitY);
    Rectangle(x1+7, SplitY+4, x2-7,
y2-7); end
  Else begin; Bar(x1+2, y1+2, x2-2, SplitY-2);
    Line(x1, SplitY, x2, SplitY); end;
end;

If (Border= 2) and (WinType <> 2) Then begin;
  Rectangle(x1+3, y1+3, x2-3, y2-3);
  Rectangle(x1+7, y1+7, x2-7, y2-7); end;
SetLineStyle(SolidLn, 0, NormWidth);
ShowCursor;
end;

PROCEDURE MemorizeLocation(j, x, y: Integer; Color: Char);
begin; With Choices[j] do begin;
  xy[1]:= x;
  xy[2]:= y;
  xy[3]:= x+TextWidth(S);
  xy[4]:= y+TextHeight(S);

  Case Color Of
(Main) 'm': begin; Fg:= White; Bg:= LightBlue; end;
(Primary) 'p': begin; Fg:= Black; Bg:= LightGray; end;
(Second) 's': begin; Fg:= Black; Bg:= Green; end;
(Tertier) 't': begin; Fg:= Black; Bg:= White; end;
(Req'd) 'r': begin; Fg:= Red; Bg:= White; end;
(Quatro) 'q': begin; Fg:= DarkGray; Bg:= LightGreen; end;
(Extra) 'e': begin; Fg:= Yellow; Bg:= Red; end;
(Boxes) 'b': begin; Fg:= Black; Bg:= LightGray; end;
(N/A) 'n': begin; Fg:= Cyan; Bg:= LightGray; end;
(Default) 'd': begin; Fg:= LightGray; Bg:= White; end;
end;
end;
end;

PROCEDURE ReadMouse;
Var pix: Integer;
x1, y1, x2, y2: Integer;
begin; j:= 0;

```

```

    pix:= 5;
    Repeat
        Delay(90);
        Getmouse(mx, my, b);
        If (Keypressed) Then begin; Getkey;
            Case Command of
                {Tab}          9: if pix < GetMaxY Then pix:= pix+5 Else pix:= GetmaxY;
                {Shft-Tab}    -43: If pix < 5 Then pix:= pix-5 Else pix:= 5;
                {Enter}       13: b:= 1;
                {Up}          14: If my > 0 Then begin; my:= my-pix; SetMouse(mx,my);
            end;
                {Right}       19: If mx < GetMaxX Then begin; mx:= mx+pix;
            SetMouse(mx,my); end;
                {Left}        17: If mx > 0 Then begin; mx:= mx-pix; SetMouse(mx,my);
            end;
                {Down}        22: If my < GetMaxY Then begin; my:= my+pix;
            SetMouse(mx,my); end;
            Else begin; Command:= 27; b:= 9; end;
        end; end;
        GetTextSettings(TextStyle);
        If TextStyle.Font = 0 Then begin; x1:= 5; x2:= 7; y1:= 1;
y2:= -1; end
            Else begin; x1:= 1; x2:= 1; y1:= 1;
y2:= 1; end;

        For i:= 1 to 255 do
            If (b= 0) and (i IN ValidChoices) and
                (mx >= Choices[i].xy[1]-x1) and (mx <=
Choices[i].xy[3]+x2) and
                (my >= Choices[i].xy[2]-y1) and (my <=
Choices[i].xy[4]+y2) Then begin;

                If (OldChoice <> i) Then begin; HideCursor; j:= 1;
                    Printxy(Choices[OldChoice].S,
Choices[OldChoice].Fg,
                        Choices[OldChoice].Bg,
                            Choices[OldChoice].xy[1],
Choices[OldChoice].xy[2]);
                    OldChoice:= i;
                    Fill.Color:= Choices[i].Bg;
                    Case Fill.Color Of
                        5: Printxy(Choices[i].S, Black,
White, Choices[i].xy[1], Choices[i].xy[2]);
                        10: Printxy(Choices[i].S, Red,
LightGray, Choices[i].xy[1], Choices[i].xy[2]);
                        15: Printxy(Choices[i].S, LightRed,
LightBlue, Choices[i].xy[1], Choices[i].xy[2]);
                        LightGray: Printxy(Choices[i].S, LightCyan,
DarkGray, Choices[i].xy[1], Choices[i].xy[2]);
                        Else Printxy(Choices[i].S, Red,
Yellow, Choices[i].xy[1], Choices[i].xy[2]);
                    end; i:= 255;
                    Showcursor; end; end

```

```

        ELSE begin; If (j= 1) and (OldChoice = i) Then begin; j:= 0;
                Hidecursor;
                Printxy(Choices[OldChoice].S,
                Choices[OldChoice].Fg, Choices[OldChoice].Bg,
                Choices[OldChoice].xy[1],
                Choices[OldChoice].xy[2]);
                OldChoice:= 0;
                Showcursor; end; end;

        Until (b = 2) or (b = 1) or (b = 4);
end;

```

END.

Unit Basics;

Interface

Uses Crt, Dos, Graph, Grmenus, IO;

```

PROCEDURE MainScreen;
PROCEDURE FindSecenek;
PROCEDURE PrintMenuItems;
PROCEDURE ReadNumber;
PROCEDURE Change(Name: AnyString);
PROCEDURE Increase;
PROCEDURE Decrease;
PROCEDURE CleanArrows;
PROCEDURE TakePicture(x1,y1,x2,y2:Integer);
PROCEDURE CleanScreen;

```

Implementation

```

PROCEDURE MainScreen;
begin; UseWin(0,0,517,25,3,1,0,LightRed,LightBlue,White,False,1);
        BarF(0,GetMaxY-20,GetMaxX,GetMaxY,Cyan);
        SetColor(LightRed);
        Rectangle(0, GetMaxY-21, GetMaxX, GetMaxY);
        BarF(528,330,632,400,LightGreen);
        SetLineStyle(SolidLn,0,ThickWidth); SetColor(Red);
        Rectangle(528,330,632,400);
        Printxy('Quick Access',Red,LightGreen,532,335);
        Line(528,356,632,356);
        For i:= 1 to 8 do

```

```

Printxy(Choices[i].S,Choices[i].Fg,Choices[i].Bg,Choices[i].xy[1],Choic
es[i].xy[2]);

```

```

        Setcolor(White);
        Line(520,0,520,GetmaxY-23);

```

```

        SetUserCharSize(1,1,1,1);

```

```

SetColor(LightBlue); SetTextStyle(1,0,4);
  OutTextxy(530,0,'ASPX');
SetColor(LightRed); SetTextStyle(1,0,1); Moveto(530,35);
  Outtext('A'); SetColor(LightCyan); OutText('ctivated');
SetColor(LightRed); Moveto(540,50);
  OutText('S'); SetColor(LightCyan); OutText('ludge');
SetColor(LightRed); Moveto(550,65);
  OutText('P'); SetColor(LightCyan); OutText('rocess');
Moveto(550,80); OutText('e');
SetColor(LightRed);
  OutText('X'); SetColor(LightCyan); OutText('pert');

SetTextStyle(0,0,0); SetColor(Cyan);
OutTextxy(530,120,'The');
OutTextxy(530,130,'Knowledge-');
OutTextxy(530,140,'Based');
OutTextxy(530,150,'Expert System');
OutTextxy(530,160,'for Activated');
OutTextxy(530,170,'Sludge');
OutTextxy(530,180,'Process');

SetColor(Cyan);
OutTextxy(530,260,'Presented by:');
SetColor(Yellow);
OutTextxy(530,275,'Naci '+Chr(153)+'zg'+Chr(129)+'r');
SetTextStyle(1,0,0);
SetUserCharSize(1,2,1,2);
end;

PROCEDURE FindSecenek;
begin; For i:= 1 to 255 do
  If (i IN ValidChoices) and
    (mx >= Choices[i].xy[1]-5) and (mx <= Choices[i].xy[3]+7)
and
    (my >= Choices[i].xy[2]-1) and (my <= Choices[i].xy[4]+1)
Then
  begin; Secenek:= i; i:= 255; end;
end;

PROCEDURE PrintMenuItems;
begin; For i:= 1 to 255 do
  If (i IN ValidChoices) Then

Printxy(Choices[i].S,Choices[i].Fg,Choices[i].Bg,Choices[i].xy[1],Choic
es[i].xy[2]);
  end;

PROCEDURE ShowError;
begin; PictureSize:= ImageSize(74,69,449,201);
MaxPictureSize:= MemAvail;
HideCursor;
If (PictureSize < MaxPictureSize) Then begin;
  GetMem(Image2,PictureSize);
  GetImage(74,69,449,201,Image2^); end

```

```

Else begin; FreeMem(Image2,PictureSize);
      GetImage(74,69,449,201,Image2^); end;
ShowCursor;
UseWin(75,70,448,200,3,1,0,Blue,Red,White,False,1);
Error:= True;
SetTextStyle(1,0,2);
Printxy('Error',White,Red,220,74);
SetTextStyle(1,0,0);
Printxy('Value entered is out of range!',Yellow,Red,150,100);
Printxy('      Valid range is:',LightCyan,Red,150,150);
Printxy('      Min= 1      Max= 14',LightCyan,Red,150,170);
ValidChoices:= [251];
PrintMenuItems;
ReadMouse;
HideCursor;
PutImage(74,69,Image2^,NormalPut);
ShowCursor;
SetTextStyle(0,0,0);
If Screen=36 Then ValidChoices:=
[ShowTrain,36,111..122,226..229, 255, 249]
      Else ValidChoices:=
[ShowTrain,37,38..110,226..229, 255, 249];
end;

```

```

PROCEDURE ReadNumber;
Label son, birdaha, bs, basildi, Enson, mm;
begin; HideCursor;
      top:= 0;
      nf:= '';

```

```

Printxy(Choices[Secenek].S,LightGray,Choices[Secenek].Bg,Choices[Secenek].xy[1],Choices[Secenek].xy[2]);
Yeri:= Choices[Secenek].xy[1]-TextWidth('W');
Delay(250);

```

```

Repeat
  Yeri:= Yeri+TextWidth(Chr(Command));
  SetColor(White);
  OutTextXY(Yeri,Choices[Secenek].xy[2],Chr(219));
bs: repeat
  SetColor(Blue);
  OutTextXY(Yeri,Choices[Secenek].xy[2],''');

  For i:= 1 to 250 do begin; Delay(1);
  Getmouse(mx,my,b);
  If b= 1 Then begin; Goto Basildi; end;
  If Keypressed Then Goto basildi; end;

  SetColor(White);
  OutTextXY(Yeri,Choices[Secenek].xy[2],''');

  For i:= 1 to 250 do begin; Delay(1);
  Getmouse(mx,my,b);
  If b= 1 Then begin; Goto Basildi; end;

```

```

        If (Keypressed) Then Goto basildi; end;

        until (Keypressed) or (b = 1);
basildi: SetColor(White);
        OutTextXY(Yeri, Choices[Secenek].xy[2], '');

        If b= 1 Then begin; b:= 0;
            Command:= 13; Goto mm; end;
        SetColor(Blue);
birdaha: Getkey;
        mm: Case Command of
            48..57::
            46: ;
            13: If top= 0 Then Goto Enson Else Goto Son;
            25: If top= 0 Then begin; XvsR(Secenek,
                Choices[Secenek].xy[1], Choices[Secenek].xy[2]);
                    Goto Enson; end;
            27: Goto Son;
            8: If top > 0 Then begin; Yeri:=
                Yeri-TextWidth(Chr(Command));
                    SetColor(White);
                    OutTextXY(Yeri, Choices[Secenek].xy[2], Chr(219));
                    top:= top-1;
                    nfs:= '';
                    For i:= 1 to top do nfs:= nfs + nf[i];
                    nf:= nfs;
                    Goto bs; end
                    Else Goto bs;
                Else goto bs;
            end;
            top:= top+1;
            if top > 6 then begin; top:= top-1; goto bs; end;
            nf:= nf + chr(Command);
            SetColor(Brown);
            OutTextxy(Yeri, Choices[Secenek].xy[2], Chr(Command));

            Until (Command = 13) or (Command = 27);
        Son: SetTextStyle(0,0,0);
            SetMouse(Choices[Secenek].xy[1], Choices[Secenek].xy[2]);
            Printxy('
', Choices[Secenek].Fg, Choices[Secenek].Bg, Choices[Secenek].xy[1], Choice
s[Secenek].xy[2]);
            Choices[Secenek].S:= '';
            For i:= 1 to (6-Length(nf)) do Choices[Secenek].S:=
                Choices[Secenek].S + ' ';
            Choices[Secenek].S:= Choices[Secenek].S + nf;
            If Secenek IN [196..199] Then
                Val(Choices[Secenek].S, Property[Secenek-195], i);
            If Secenek IN [192..195] Then
                Val(Choices[Secenek].S, State[Secenek-191], i);

        {pH}      If Secenek IN [53..58] Then begin;
                Val(Choices[Secenek].S, pH, i);

```

```

                If (pH < 1) OR (pH > 14) Then begin;
ShowCursor;
                ShowError;

XvsR(Secenek, Choices[Secenek].xy[1], Choices[Secenek].xy[2]);

Printxy(Choices[Secenek].S, Choices[Secenek].Fg, Choices[Secenek].Bg,
Choices[Secenek].xy[1], Choices[Secenek].xy[2]);
                HideCursor; end; end;

    Enson:
Printxy(Choices[Secenek].S, Choices[Secenek].Fg, Choices[Secenek].Bg, Choices[Secenek].xy[1], Choices[Secenek].xy[2]);
                ShowCursor;
                If (Secenek < 187) and (Secenek < 247) Then
DBRecord[1].Values[Secenek]:= Choices[Secenek].S;
                If Secenek IN [111..112] Then
MemorizeLocation(Secenek, 100+68*(Secenek-111), 201, 'r');
                If Secenek IN [121..122] Then
MemorizeLocation(Secenek, 372+68*(Secenek-121), 239, 'r');
                end;

PROCEDURE Change(Name: AnyString);
    begin; If Altered Then WriteDataBase;
                If Name= 'Date' Then Case Secenek Of
                        226: ValidChoices:=
[220..222, 227, 228];
                        227: ValidChoices:=
[220..222, 226, 228];
                        228: ValidChoices:=
[220..222, 226, 227];
                                end
                        Else ValidChoices:= [223..225];
                PrintMenuItems;

Printxy(Choices[Secenek].S, Black, LightGray, Choices[Secenek].xy[1], Choices[Secenek].xy[2]);
                Oldchosen:= Secenek;
    end;

PROCEDURE Increase;
    begin; Val(Choices[OldChosen].S, k, j);
                Case OldChosen Of
                        226: If k < 12 Then k:= k+1 Else k:= 12;
                        227: If k < 31 Then k:= k+1 Else k:= 31;
                        228: If k < 92 Then k:= k+1 Else k:= 92;
                        229: If k < 20 Then k:= k+8
                                Else k:= 20;
                end;
                If k < 10 Then begin; Str(k, Choices[OldChosen].S);
                        Choices[OldChosen].S:=
'0'+Choices[OldChosen].S; end
                        Else Str(k, Choices[OldChosen].S);

```



```

Printxy(Choices[OldChosen].S,Black,LightGray,Choices[OldChosen].xy[1],C
hoices[OldChosen].xy[2]);
    Delay(100);
end;

PROCEDURE Decrease;
begin; Val(Choices[OldChosen].S,k,j);
    Case OldChosen Of
        226: If k > 1 Then k:= k-1 Else k:= 1;
        227: If k > 1 Then k:= k-1 Else k:= 1;
        228: If k > 86 Then k:= k-1 Else k:= 86;
        229: If k > 4 Then k:= k-8 Else k:= 4;
    end;
    If k < 10 Then begin; Str(k,Choices[OldChosen].S);
        Choices[OldChosen].S:=
'0'+Choices[OldChosen].S; end
        Else Str(k,Choices[OldChosen].S);

Printxy(Choices[OldChosen].S,Black,LightGray,Choices[OldChosen].xy[1],C
hoices[OldChosen].xy[2]);
    Delay(100);
end;

PROCEDURE CleanArrows;
begin; If Screen= 37 Then ValidChoices:= [37,38..110,226..229, 255]
        Else ValidChoices:= [36,111..122,226..229,
255];
    Printxy(Choices[OldChosen].S,Choices[OldChosen].Fg,
Choices[OldChosen].Bg,Choices[OldChosen].xy[1],Choices[OldChosen].xy[2]
);
    BarF(235,31,282,56,White);
    BarF(443,31,492,56,White);
    ReadDataBase;
end;

PROCEDURE TakePicture(x1,y1,x2,y2:Integer);
begin; PictureSize:= ImageSize(x1,y1,x2,y2);
    MaxPictureSize:= MemAvail;
    HideCursor;
    If (PictureSize < MaxPictureSize) Then begin;
        GetMem(Image,PictureSize);
        GetImage(x1,y1,x2,y2,Image^); end
    Else begin; FreeMem(Image,PictureSize);
        GetImage(x1,y1,x2,y2,Image^); end;
    ShowCursor;
end;

PROCEDURE CleanScreen;
begin; Clean(0,28,518,GetMaxY-24); Fill.Color:= LightBlue;
SetTextStyle(1,0,0);
    If (MenuOut <= 3) OR (MenuOut = 99)
        Then begin; ValidChoices:= [1..8];
            PrintMenuItems; end

```

```

        end;
        Month:= Month - 1;
Until Month <= 1;
Val(Choices[227].S,Day,i);
top:= top+(Day-1)*3;
Val(Choices[229].S,Hour,i);
Case Hour Of
    4: top:= top+1;
    12: top:= top+2;
    20: top:= top+3;
end;
RecordNumber:= top-1;    (Actual Record starts from #0)
end;

PROCEDURE ReadDataBase;
Label Cont, Skip;
begin; Yil:= '19'+Choices[228].S;
Assign(DBFile,Yil+'.DB');
($I-) Reset(DBFile); ($I+)
If IOResult < 0 Then begin; Ok:= False; Goto Skip; end;
FindRecordNumber;
Seek(DBFile,RecordNumber);
($I-) Read(DBFile,DBRecord[1]); ($I+)
If IOResult= 0 Then Ok:= True Else Ok:= False;
For i:= 2 to 3 do begin; Seek(DBFile,RecordNumber-i+1);
Read(DBFile,DBRecord[i]); end;

Skip: Required:=
[41,42,45,46,47,51,52,53,57,58,62,80,81,82,87,88,91,92,99,
102,103,104,104,105,106,107,108,111,112,113,114,117,118,119,
120,121,122];
k:= 37;
For j:= 1 to 15 do
    For i:= 1 to 6 do begin;
        k:= k + 1;
        If k >= 111 Then Goto Cont;
        Case k Of
            41,43,45,109: i:= i + 1;
            90: i:= i+2;
            105,107: i:= i+4;
        end;

        If Ok Then begin; Choices[k].S:= DBRecord[1].Values[k];
            If (Choices[k].S= 'Req'd ') OR
                (k IN Required) Then
MemorizeLocation(k,100+68*(i-1),87+(j-1)*19,'r')
            Else
MemorizeLocation(k,100+68*(i-1),87+(j-1)*19,'t'); end
            Else XvsR(k,100+68*(i-1),87+(j-1)*19); end;
    Cont: k:= 110;
    For j:= 1 to 3 do
        For i:= 1 to 2 do begin;
            k:= k + 1;

```

```

        If Ok Then begin; Choices[k].S:= DBRecord[1].Values[k];
          If (Choices[k].S= 'Req'd ') OR
            (k IN Required) Then
MemorizeLocation(k,100+68*(i-1),201+(j-1)*19,'r')
          Else
MemorizeLocation(k,100+68*(i-1),201+(j-1)*19,'t'); end
          Else XvsR(k,100+68*(i-1),201+(j-1)*19); end;
      k:= 116;
      For j:= 1 to 3 do
        For i:= 1 to 2 do begin;
          k:= k + 1;

          If Ok Then begin; Choices[k].S:= DBRecord[1].Values[k];
            If (Choices[k].S= 'Req'd ') OR
              (k IN Required) Then
MemorizeLocation(k,372+68*(i-1),201+(j-1)*19,'r')
            Else
MemorizeLocation(k,372+68*(i-1),201+(j-1)*19,'t'); end
            Else XvsR(k,372+68*(i-1),201+(j-1)*19); end;
          If Ok Then Close(DBFile);
        end;
end;

```

```

PROCEDURE WriteDataBase;
begin; Yil:= '19'+Choices[228].S;
      kk:= Train-230;
      Assign(DBFile,Yil+'.DB'); Reset(DBFile);
      FindRecordNumber;
      Seek(DBFile,RecordNumber);
      Write(DBFile,DBRecord[1]);
      Close(DBFile);
      Assign(dt,'NH4-Sim.in'); Rewrite(dt);
      Writeln(dt,Choices[41].S);
      Writeln(dt,Choices[102].S);
      Writeln(dt,Choices[103].S);

      Ok:= True;
      X:= GetValue(111+kk);    QW:= GetValue(119+kk);
      XW:= GetValue(121+kk);  Q := GetValue( 41+kk);
      XE:= GetValue( 87+kk);
      If (Ok) and (((QW*XW)+(Q*XE)) <> 0) Then begin;
        XSA:= (900000*X)/((QW*XW)+(Q*XE))/24/60;
        If XSA > 30 Then Writeln(dt,'30')
          Else Writeln(dt,XSA:6:1); end;

      Writeln(dt,Choices[57].S);
      Writeln(dt,Choices[113].S);
      Writeln(dt,Choices[51].S);
      Close(dt);
      Altered:= False;
end;

```

```

Function GetValue(n: Integer): Real;
begin; If (Choices[n].S = ' X ') OR
      (Choices[n].S = 'Req'd ') Then Ok:= False

```

```

Else begin;
Val(Choices[n].S,Null,i);
Getvalue:= Null;
end;
end;

PROCEDURE WriteParameterValues;
Procedure WriteValue(n: Integer);
begin; If (Choices[n].S = ' X ') OR
(Choices[n].S = 'Req'd ') Then Writeln(dt,'Missing')
Else Writeln(dt,Choices[n].S);
end;
begin; kk:= Train-230;
Assign(dt,'INPUT.VAL'); Rewrite(dt);
WriteValue( 41+kk); {Q}
WriteValue( 80 ); {CODin}
WriteValue( 81+kk); {CODout}
WriteValue( 87+kk); {Xe}
WriteValue(111+kk); {X}
WriteValue(119+kk); {Qw}
WriteValue(121+kk); {Xw}
WriteValue( 45+kk); {Qr}
WriteValue(117+kk); {SBD}
Writeln(dt,'900000'); {V}
WriteValue( 51+kk); {Temp}
WriteValue( 47 ); {ITemp}
WriteValue( 91+kk); {PO4 Out}
WriteValue(113+kk); {DO}
WriteValue( 57+kk); {pH}
WriteValue( 53 ); {IpH}
WriteValue( 62 ); {Sulfide}
WriteValue( 99 ); {INH4}
WriteValue(102 ); {NH4 Eq}
WriteValue(103+kk); {NH4 Out}
WriteValue(105+kk); {NO2 Out}
WriteValue(107+kk); {NO3 Out}

Ok:= True;
X:= GetValue(111+kk); QW:= GetValue(119+kk);
XW:= GetValue(121+kk); Q := GetValue( 41+kk);
XE:= GetValue( 87+kk);
If (Ok) and (((QW*XW)+(Q*XE)) <> 0) Then begin;
XSA:= (900000*X)/((QW*XW)+(Q*XE))/24/60;
If XSA > 30 Then Writeln(dt,'"Sludge Age calculation is
not reliable"')
Else Writeln(dt,XSA:6:1); end
Else Writeln(dt,'"Cannot Calculate"');

Ok:= True;
X:= GetValue(111+kk); COD:= GetValue(80);
Q:= GetValue( 41+kk);
If X= 0 Then Ok:= False;
If Ok Then begin; FM:= (COD*0.6*Q*60*24)/(X*1200000);
Writeln(dt,FM:6:3); end

```

```

Else Writeln(dt, 'Cannot Calculate');

Ok:= True;
COD:= GetValue(80);      Q := GetValue(41+kk);
If Ok Then begin; ORGLOAD:= (COD*0.6*Q*60*24*3.7854/1000000);
Writeln(dt, ORGLOAD:6:0);      EPO4:= (ORGLOAD / 100);
Writeln(dt, EPO4:6:1); end
      Else begin; Writeln(dt, 'Organic loading is not
available');
      Writeln(dt, 'Estimated PO4 is not
available'); end;

Ok:= True;
X:= GetValue(111+kk);    XE:= GetValue(87+kk);
XW:= GetValue(121+kk);   Q := GetValue(41+kk);
If Q= 0 Then Ok:= False;
If Ok Then begin; OPTQW:=
(1200000*X)/(20*XW*24*60)-((Q*XE)/XW); Writeln(dt, OPTQW:6:1);
      OPTRAS:= (((X*Q)-(XW*OPTQW))/(XW-X));
Writeln(dt, OPTRAS:6:1);
      OPTRASP:= ((OPTRAS/Q)*100);
Writeln(dt, OPTRASP:6:0); end
      Else begin; Writeln(dt, 'Optimum WAS Flowrate is not
available');
      Writeln(dt, 'Optimum RAS Flowrate is not
available');
      Writeln(dt, 'Optimum RAS % is not
available'); end;

Ok:= True;
Q := GetValue(41+kk);    QR:= GetValue(45+kk);
If Q= 0 Then Ok:= False;
If Ok Then begin; CurRASP:= ((QR/Q)*100);
Writeln(dt, CurRASP:6:0); end
      Else Writeln(dt, '%RAS Flowrate is not available');

Close(dt);
end;

PROCEDURE WriteSearchTech;
begin; Assign(dt, 'SEARCH.TCH'); Rewrite(dt);
Case SearchTech Of
  232: begin; Writeln(dt, 'Limited');
        Writeln(dt, 'NO'); end;
  233: begin; Writeln(dt, 'Regular');
        Writeln(dt, 'NO'); end;
  234: begin; Writeln(dt, 'Indepth');
        Writeln(dt, 'YES'); end;
end;
Close(dt);
end;
END.

```

Unit Plot;

Interface

Uses Graph, Grmenus;

```
PROCEDURE XX(Func: Char; n: Word);
PROCEDURE MemorizeSquares(n1, n2, x1, y1: Integer);
PROCEDURE CalcStats(n: Integer);
PROCEDURE DrawSquareX;
PROCEDURE SetPatternAnalysis;
PROCEDURE SetDBStatistics;
PROCEDURE PlotGraph(C: Char);
PROCEDURE Xaxis(Period: Integer);
PROCEDURE Yaxis(Hangi, LimitL, LimitH: Integer);
```

Implementation

Var MaxTop: Word;

```
PROCEDURE XX(Func: Char; n: Word);
  begin; Case Func Of
    (Draw)  'D': begin; SetColor(Red);
              Line(Squares[n].xy[1], Squares[n].xy[2],
                  Squares[n].xy[3], Squares[n].xy[4]);
              Line(Squares[n].xy[3], Squares[n].xy[2],
                  Squares[n].xy[1], Squares[n].xy[4]); end;
    (Erase) 'E': BarF(Squares[n].xy[1]+1, Squares[n].xy[2]+1,
                    Squares[n].xy[3]-1, Squares[n].xy[4]-1, White);
  end;
end;
```

```
PROCEDURE MemorizeSquares(n1, n2, x1, y1: Integer);
  begin; For i:= 1 to n2-n1+1 do begin;
    Rectangle(x1,y1+(i-1)*17,x1+15,y1+13+(i-1)*17);
    With Squares[n1+i-1] do begin;
      xy[1]:= x1;
      xy[2]:= y1+(i-1)*17;
      xy[3]:= x1+15;
      xy[4]:= y1+13+(i-1)*17;
    end; end;
  end;
```

```
PROCEDURE DrawSquareX;
  begin; For j:= 1 to 100 do
    If (mx >= Squares[j].xy[1]) and (mx <= Squares[j].xy[3])
  and
    (my >= Squares[j].xy[2]) and (my <= Squares[j].xy[4])
  and
    (j IN ValidSquares) Then begin; HideCursor;
  Case j Of
    1..21: If Xed[j]= False Then begin; If top < MaxTop
  Then
```

```

begin; XX('D',j);
Xed[j]:= True; top:= top+1;end; end
Else begin; XX('E',j);
Xed[j]:= False; top:= top-1; end;
(Location) 22..29: For i:= 22 to 29 do If i <> j Then begin;
XX('E',i);
Xed[i]:=
False; end
Else begin;
XX('D',i); DefLocation:= i;
Xed[i]:=
True; end;
(Duration) 30..34: For i:= 30 to 34 do If i <> j Then begin;
XX('E',i);
Xed[i]:=
False; end
Else begin;
XX('D',i); DefDuration:= i;
Xed[i]:=
True; end;
35;;
(Type of Plot) 36..38: For i:= 36 to 38 do If i <> j Then begin;
XX('E',i);
Xed[i]:=
False; end
Else begin;
XX('D',i); DefType:= i;
SetTextStyle(0,0,0); top:= 0;
Case i Of
36: begin;
Printxy(' 5 Parameters',Red,White,10,440);
MaxTop:= 5; end;
37: begin;
Printxy(' 2 Parameters',Red,White,10,440);
MaxTop:= 2; end;
38: begin;
Printxy(' 1 Parameter ',Red,White,10,440);
MaxTop:= 1; end;
end;
For k:= 1
to 21 do if (Xed[k]= True) and (top < MaxTop)
Then top:= top+1
Else begin; Xed[k]:= False; XX('E',k); end;
SetTextStyle(1,0,0);
Xed[i]:=
True; end;

```

```

(Open)          39..40;;
{ASP 600}       41: begin; Train:= 230; XX('E',42); XX('D',41);
end;
{ASP 700}       42: begin; Train:= 231; XX('E',41); XX('D',42);
end;
{SearchTech}   43..45: For i:= 43 to 45 do If i <> j Then begin;
XX('E',i);
False; end
XX('D',i); SearchTech:= i+189;
True; end;
{DB Stats}     51..71: For i:= 51 to 71 do If i <> j Then begin;
XX('E',i);
False; end
XX('D',i); CalcStats(i);
True; end;
72..74: For i:= 72 to 74 do If i <> j Then begin;
XX('E',i);
False; end
XX('D',i);
True; end;
75..76: If Xed[j]= False Then begin; XX('D',j);
Xed[j]:= True; end
Else begin; XX('E',j);
Xed[j]:= False; end;
end;
ShowCursor; end;
end;

```

PROCEDURE SetPatternAnalysis;

```

begin; Printxy('Raw Feed',Black,White,210,73);
Printxy('DAF T-202 Effluent',Black,White,210,90);
Printxy('DAF T-302 Effluent',Black,White,210,107);
Printxy('Equalization T-500',Black,White,210,124);
Printxy('ASP T-600 Basin',Black,White,210,141);
Printxy('ASP T-700 Basin',Black,White,210,158);
Printxy('Clarifier T-601 Effluent',Black,White,210,175);
Printxy('Clarifier T-701 Effluent',Black,White,210,192);

Printxy('Daily',Black,White,390,73);
Printxy('Weekly',Black,White,390,90);
Printxy('Monthly',Black,White,390,107);
Printxy('Yearly',Black,White,390,124);
Printxy('Specific Dates',Black,White,390,141);

```



```

Printxy('Pattern Analysis',Black,White,320,293);
Printxy('Variable Interrelations',Black,White,320,310);
Printxy('Parameter Tracking',Black,White,320,327);

Printxy('Ammonia',Blue,White,28,73);
Printxy('Dissolved Oxygen',Blue,White,28,90);
Printxy('Flowrate',Blue,White,28,107);
Printxy('IR',Blue,White,28,124);
Printxy('MLSS',Blue,White,28,141);
Printxy('Nitrate',Blue,White,28,158);
Printxy('Nitrite',Blue,White,28,175);
Printxy('pH',Blue,White,28,192);
Printxy('Phenol',Blue,White,28,209);
Printxy('Phosphate',Blue,White,28,226);
Printxy('Polymer feed',Blue,White,28,243);
Printxy('RAS Flowrate',Blue,White,28,260);
Printxy('Sludge Depth',Blue,White,28,277);
Printxy('Sludge MLSS',Blue,White,28,294);
Printxy('Sulfide',Blue,White,28,311);
Printxy('Sulfite',Blue,White,28,328);
Printxy('SVI',Blue,White,28,345);
Printxy('Temperature',Blue,White,28,362);
Printxy('TOC',Blue,White,28,379);
Printxy('TSS',Blue,White,28,396);
Printxy('Waste Flowrate',Blue,White,28,413);
Printxy('Choose Maximum of',Red,White,10,430);
Printxy(' 5 Parameters',Red,White,10,440);
top:= 0; MaxTop:= 5;

end;

PROCEDURE CalcStats(n: Integer);
begin; Case n Of
51: begin; Printxy('85',Red,Cyan,405,113);
Printxy('75',Red,Cyan,405,133);
Printxy('70',Red,Cyan,405,153);
Printxy('Ammonia > 100',Red,Cyan,295,213);
Printxy('3',Red,Cyan,335,320); end;
58: begin; Printxy('5.7',Red,Cyan,405,113);
Printxy('6.9',Red,Cyan,405,133);
Printxy('7.0',Red,Cyan,405,153);
Printxy('6.5 < pH < 7.3',Red,Cyan,295,213);
Printxy('7',Red,Cyan,335,320); end;

end;

end;

PROCEDURE SetDBStatistics;
begin; Printxy('Ammonia',Blue,White,28,73);
Printxy('Dissolved Oxygen',Blue,White,28,90);
Printxy('Flowrate',Blue,White,28,107);
Printxy('IR',Blue,White,28,124);
Printxy('MLSS',Blue,White,28,141);
Printxy('Nitrate',Blue,White,28,158);
Printxy('Nitrite',Blue,White,28,175);
Printxy('pH',Blue,White,28,192);

```

```

Printxy('Phenol',Blue,White,28,209);
Printxy('Phosphate',Blue,White,28,226);
Printxy('Polymer feed',Blue,White,28,243);
Printxy('RAS Flowrate',Blue,White,28,260);
Printxy('Sludge Depth',Blue,White,28,277);
Printxy('Sludge MLSS',Blue,White,28,294);
Printxy('Sulfide',Blue,White,28,311);
Printxy('Sulfite',Blue,White,28,328);
Printxy('SVI',Blue,White,28,345);
Printxy('Temperature',Blue,White,28,362);
Printxy('TOC',Blue,White,28,379);
Printxy('TSS',Blue,White,28,396);
Printxy('Waste Flowrate',Blue,White,28,413);

```

```

BarF(240,105,470,370,Cyan);
SetLineStyle(SolidLn,0,ThickWidth); SetColor(Red);
Rectangle(240,105,470,370);
Printxy('    Current Value = ',White,Cyan,250,113);
Printxy('    Typical Value = ',White,Cyan,250,133);
Printxy('Value desired Now = ',White,Cyan,250,153);
Printxy('Value considered "abnormal"',White,Cyan,250,193);
Printxy('Number of times the Value',White,Cyan,250,260);
Printxy('was considered "abnormal"',White,Cyan,250,280);
Printxy('within the last 3 days',White,Cyan,250,300);
SetLineStyle(SolidLn,0,NormalWidth);

```

end;

PROCEDURE ReadParameters;

```

begin; For j:= 1 to 5 do For i:= 1 to 93 do Parameter[j,i]:= 0;
Max:= 0;
j:= 0;
Case DefDuration Of
  30: n:= 3;  31: n:= 21;  32: n:= 84;
end;
For k:= 1 to 21 do If Xed[k] Then begin; j:= Succ(j);
Case k Of
  1: FileName:= '1988-NH3';  2: FileName:= '1988-DO';
  3: FileName:= '1988-QIN';  4: FileName:= '1988-IR';
  5: FileName:= '1988-MLS';  6: FileName:= '1988-NO3';
  7: FileName:= '1988-NO2';  8: FileName:= '1988-PH';
  9: FileName:= '1988-PHL'; 10: FileName:= '1988-PO4';
 11: FileName:= '1988-PLY'; 12: FileName:= '1988-QR';
 13: FileName:= '1988-SBD'; 14: FileName:= '1988-XW';
 15: FileName:= '1988-SO3'; 16: FileName:= '1988-S';
 17: FileName:= '1988-SVI'; 18: FileName:= '1988-TMP';
 19: FileName:= '1988-COD'; 20: FileName:= '1988-TSS';
 21: FileName:= '1988-QW';
end;

Assign(IndDB[j],FileName+'.DB'); Reset(IndDB[j]);
For i:= 1 to n do begin;
  Seek(IndDB[j],RecordNumber-i+1);
  Read(IndDB[j],IndRecord);
  Val(IndRecord.Values[DefLocation-21],Parameter[j,i],ii);

```

```

        If Parameter[j,i] > Max Then Max:=
Round(Parameter[j,i]); end;
        Close(IndDB[j]); end;
        Str(Max:5,Choices[247].S); Choices[247].S:= Choices[247].S+'
';
    end;

PROCEDURE DrawParameter;
begin; For k:= 1 to j do begin;
    MoveTo(60,Trunc(420-Parameter[k,1]*(420-97)/Max));
    SetColor(k+8);
    For i:= 1 to n do
        If Parameter[k,i] > 0 Then
OutTextXY(60+Trunc((i-1)*390/(n-1)),
            Trunc(420-Parameter[k,i]*(420-97)/Max),'*'); end;
    end;

PROCEDURE PlotGraph(C: Char);
Label WrongData, Pass;
var xxx, yyy: Integer;
begin; SetLineStyle(SolidLn,0,ThickWidth);    SetColor(Blue);
    Line(60,80,60,420);
    Line(60,420,480,420);
    If (C = 'T') or (C = 'I') Then Goto Pass;
    SetTextJustify(CenterText,TopText);
    Axis(DefDuration);
    ReadParameters;
    If Max = 0 Then Goto WrongData;
    Yaxis(247,Max,20000);
Pass:    Case C Of
        'P': begin;
            SetLineStyle(SolidLn,0,NormWidth);
            DrawParameter;

            ValidChoices:= [247,255];
            end;
        'I':begin; SetTextStyle(0,0,0);
            Printxy('Dissolved Oxygen
Concentration',DarkGray,LightGray,150,430);
            SetTextStyle(0,1,0);
            OutTextxy(50,200,'Air Pressure Gauge Reading');
            SetTextStyle(0,0,0);
            SetLineStyle(SolidLn,0,NormWidth);    SetColor(Red);
            Randomize;
            xxx:= 60;
            yyy:= 400;
            MoveTo(xxx,yyy);
            For i:= 1 to 7 do begin; xxx:= xxx+(5+Random(80));
                yyy:= yyy-(5+Random(40));
                If yyy < 60 Then yyy:= 60;
                LineTo(xxx,yyy); end; end;
        'T':begin; SetTextStyle(0,0,0);
            Line(60,30,60,420);
            Line(60,420,500,420);

```

```

Printxy('COD Tracking',DarkGray,LightGray,250,430);
SetTextStyle(0,1,0);
OutTextxy(50,150,'COD Concentration (mg/L)');
SetTextStyle(0,0,0);
Line(80,100,155,100); Line(155,100,172,70);
Line(155,100,172,130);
Line(212,70,230,100); Line(212,130,230,100);
Circle(192,70,20); Circle(192,130,20);
Line(230,100,250,100); Rectangle(250,50,285,150);
Line(285,100,305,100);
Line(315,75,305,75); Line(305,75,305,125);
Line(305,125,315,125);
Rectangle(315,60,370,90); Rectangle(315,110,370,140);
Line(370,75,380,75); Line(380,75,388,60);
Line(370,125,380,125); Line(380,125,388,140);
Circle(418,65,30); Circle(418,135,30);
Line(448,65,470,65); Line(470,65,470,135);
Line(470,135,448,135);
Line(470,100,500,100);
SetTextJustify(1,1); SetColor(Red);
OutTextxy(192,70,'200');
OutTextxy(192,130,'300');
OutTextxy(268,100,'500');
OutTextxy(343,75,'T-600');
OutTextxy(343,125,'T-700');
OutTextxy(418,65,'T-601');
OutTextxy(418,135,'T-701');

SetLineStyle(DottedLn,0,NormWidth); SetColor(White);
Line(155,30,155,420); Line(230,30,230,420);
Line(305,30,305,420); Line(380,30,380,420);
Line(455,30,455,420);
SetLineStyle(SolidLn,0,ThickWidth);
SetColor(LightGreen);

Line(80,150,155,150); MoveTo(155,150); LineTo(230,200);
LineTo(305,230);
LineTo(380,370); LineTo(455,390); LineTo(500,390);
SetColor(DarkGray);
OutTextxy(100,150,'1,500');
OutTextxy(230,200,'1,210');
OutTextxy(305,230,'1,036');
OutTextxy(380,370,'224');
OutTextxy(455,390,'108');
SetTextJustify(0,2); end;
end;
WrongData:SetTextJustify(LeftText,TopText);
end;

PROCEDURE XTics(n: Integer);
begin; SetLineStyle(SolidLn,0,NormWidth); SetColor(Blue);
For i:= 1 to n do
Line(60+Trunc((i-1)*390/(n-1)),420,60+Trunc((i-1)*390/(n-1)),423);
end;

```

```

PROCEDURE Xaxis(Period: Integer);
begin; Case Period Of
    30: begin; SetTextStyle(0,0,0);
        Printxy('4',White,LightGray, 60,425);
        Printxy('12',White,LightGray,255,425);
        Printxy('20',White,LightGray,450,425);
        SetTextJustify(LeftText,TopText);
        Printxy('Daily Pattern
Analysis',DarkGray,LightGray,150,440);
        XTics(3);
        SetTextStyle(0,1,0);
        OutTextxy(20,200,'Parameters');
        SetTextStyle(0,0,0); end;
    31: begin; SetTextStyle(0,0,0);
        Printxy('Mon',White,LightGray, 60,425);
        Printxy('Tue',White,LightGray,125,425);
        Printxy('Wed',White,LightGray,190,425);
        Printxy('Thu',White,LightGray,255,425);
        Printxy('Fri',White,LightGray,320,425);
        Printxy('Sat',White,LightGray,385,425);
        Printxy('Sun',White,LightGray,450,425);
        SetTextJustify(LeftText,TopText);
        Printxy('Weekly Pattern
Analysis',DarkGray,LightGray,150,440);
        XTics(7);
        SetTextStyle(0,1,0);
        OutTextxy(50,200,'Parameters');
        SetTextStyle(0,0,0); end;
    32: begin; SetTextStyle(0,0,0);
        Printxy('1',White,LightGray,60,425);
        For i:= 1 to 6 do begin; Str(i*5,labelit);
            Printxy(Labelit,White,LightGray,112+(i-1)*65,425);
        end;
        SetTextJustify(LeftText,TopText);
        Printxy('Monthly Pattern
Analysis',DarkGray,LightGray,150,440);
        XTics(31);
        SetTextStyle(0,1,0);
        OutTextxy(50,200,'Parameters');
        SetTextStyle(0,0,0); end;
end;
end;

PROCEDURE Yaxis(Hangi, LimitL, LimitH: Integer);
begin; Val(Choices[Hangi].S,Max,i);
    If Max < LimitL Then Max:= LimitL;
    If Max > LimitH Then Max:= LimitH;
    If Hangi= 187 Then For i:= 1 to 3 do If P[i] > Max Then
P[i]:= Max-i;
    Case Max of
        1.. 20: tic:= 1;
        21.. 50: tic:= 5;
        51..150: tic:= 10;
        151..250: tic:= 20;

```

```

        251..500: tic:= 50;
        501..1000: tic:= 100;
        1001..5000: tic:= 200;
        Else tic:= 500;
        end;
        If Hangi= 187 Then BarF(25,90,58,420,White)
            Else BarF(25,90,58,420,LightGray);
        Tane:= (Max DIV tic);
        SetTextStyle(0,0,0);
        SetTextJustify(RightText,CenterText);
        SetLineStyle(SolidLn,0,NormWidth);
        SetColor(Blue);
        OutTextxy(53,420,' 0');

        For i:= 1 to tane do begin;
            Str((i*tic):5,labelit);
            Line(55,Trunc(420-i*(320/tane)),
60,Trunc(420-i*(320/tane)));
            OutTextxy(53,Trunc(420-i*(320/tane)),labelit); end;

        Tane:= 2*Tane;
        For i:= 1 to tane do
            Line(57,Trunc(420-i*(320/tane)),
60,Trunc(420-i*(320/tane)));

        Line(55,100,60,100);
        SetTextJustify(LeftText,TopText);
        MemorizeLocation(188,385,Trunc(416-(Std*320/Max)),'e');
        For i:= 1 to 3 do

MemorizeLocation(188+i,381,Trunc(416-(P[i]*320/Max)),'t');
        NewMax:= False;
        end;
END.

```

Unit MenusEXT;

Interface

Uses Crt, Dos, Graph, Grmenus, IO, Basics, Plot, States, SA;

```

PROCEDURE Beginning;
PROCEDURE PrimaryMenu(i: Byte);
PROCEDURE SecondaryMenu(i: Byte; S:AnyString);
PROCEDURE Confirmation;
PROCEDURE SetLimits;
PROCEDURE SetupDataEntery;
PROCEDURE MainTrain;
PROCEDURE ASPTrain;
PROCEDURE UpdateCalcs;
PROCEDURE SetConcStatements;

```

Implementation

```
var emergency: byte;
    Problem, Missing, UK, NotAsked, Okay: Set Of 1..101;
    Cat: Array [1..5,0..50] Of Integer;

PROCEDURE Beginning;
begin; SetTextStyle(1,0,0);
    SetUserCharSize(1,2,1,2);
    Choices[0].S:= '';
    MemorizeLocation(0,0,0,'m');
    Choices[1].S:= ' Expert Systems ';
    MemorizeLocation(1,7,4,'m');
    Choices[2].S:= ' Data Base Functions';
    MemorizeLocation(2,147,4,'m');
    Choices[3].S:= ' Generate Reports';
    MemorizeLocation(3,324,4,'m');
    Choices[4].S:= ' Quit';
    MemorizeLocation(4,470,4,'m');
    Choices[5].S:= ' Help ';
    MemorizeLocation(5,200,459,'e');
    Choices[6].S:= ' Glossary ';
    MemorizeLocation(6,30,459,'e');

    Choices[7].S:= 'DB Records';
    MemorizeLocation(7,538,360,'q');
    Choices[8].S:= 'Run ASPX ';
    MemorizeLocation(8,538,380,'q');

    Choices[15].S:= ' Preliminary Report ';
    MemorizeLocation(15,296,62,'p');
    Choices[16].S:= ' Daily Report ';
    MemorizeLocation(16,296,87,'p');
    Choices[17].S:= ' Operator Activity Report';
    MemorizeLocation(17,296,112,'p');
    Choices[18].S:= ' Change Report Format';
    MemorizeLocation(18,296,137,'p');

    Choices[19].S:= ' Run Diagnostic Consultation ';
    MemorizeLocation(19,25,62,'p');
    Choices[20].S:= ' Display Nitrification State ';
    MemorizeLocation(20,25,85,'p');
    Choices[21].S:= ' Run ASPX Mathematical Model ';
    MemorizeLocation(21,25,107,'p');
    Choices[22].S:= ' Set Parameter Limits ';
    MemorizeLocation(22,25,147,'p');
    Choices[23].S:= ' Define Nitrification States ';
    MemorizeLocation(23,25,167,'p');
    Choices[24].S:= ' Modify Conclusion Statements ';
    MemorizeLocation(24,25,187,'p');

    Choices[25].S:= ' Add/Change Data Records ';
    MemorizeLocation(25,111,62,'p');
    Choices[26].S:= ' Pattern Analysis/Graphics';
```

```

MemorizeLocation(26,111,85,'p');
Choices[27].S:= ' Data Base Statistics      ';
MemorizeLocation(27,111,107,'p');
Choices[28].S:= ' Transfer Data Base Files  ';
MemorizeLocation(28,111,129,'p');
Choices[29].S:= ' Confirm Data Base Records';
MemorizeLocation(29,111,151,'p');

Choices[30].S:= ' Logout : End ASPX  ';
MemorizeLocation(30,180,200,'t');
Choices[31].S:= ' Exit : Goto DOS  ';
MemorizeLocation(31,195,230,'t');

Choices[32].S:= ' Run  ';
MemorizeLocation(32,230,177,'m');

Choices[33].S:= ' OK  ';
MemorizeLocation(33,230,227,'m');

Choices[34].S:= ' ASP600  ';
MemorizeLocation(34,21,391,'m');
Choices[35].S:= ' ASP700  ';
MemorizeLocation(35,21,391,'m');
Choices[36].S:= ' Select: Feed, DAF200 & 300, EQ500  ';
MemorizeLocation(36,130,385,'m');
Choices[37].S:= ' Select: ASP600 & 700, ST601 & 701  ';
MemorizeLocation(37,130,385,'m');

ReadStates;

Choices[184].S:= ' Don''t erase any names  ';

Choices[185].S:= ' No  ';
MemorizeLocation(185,170,210,'s');
Choices[186].S:= ' Yes  ';
MemorizeLocation(186,280,210,'s');

Choices[200].S:= 'Add a name';
MemorizeLocation(200,75,180,'s');
Choices[201].S:= 'Erase a name';
MemorizeLocation(201,325,180,'s');

Choices[213].S:= 'Try Again';
MemorizeLocation(213,170,280,'q');
Choices[214].S:= 'Exit';
MemorizeLocation(214,280,280,'q');

Choices[215].S:= ' Severe Problem  ';
MemorizeLocation(215,50,405,'t');
Choices[216].S:= ' Problem  ';
MemorizeLocation(216,50,423,'t');
Choices[217].S:= 'Potential Problem';
MemorizeLocation(217,50,440,'t');
Choices[218].S:= ' <<  ';

```



```

MemorizeLocation(218,270,423,'t');
Choices[219].S:= ' >> ';
MemorizeLocation(219,340,423,'t');

Choices[220].S:= ' '+Chr(30)+' ';
MemorizeLocation(220,240,32,'p');
Choices[221].S:= ' '+Chr(31)+' ';
MemorizeLocation(221,240,47,'p');
Choices[222].S:= Chr(177);
MemorizeLocation(222,272,40,'p');

Choices[223].S:= ' '+Chr(30)+' ';
MemorizeLocation(223,450,32,'p');
Choices[224].S:= ' '+Chr(31)+' ';
MemorizeLocation(224,450,47,'p');
Choices[225].S:= Chr(177);
MemorizeLocation(225,482,40,'p');

GetDate(Year,Month,Day,DayofWeek);
If Month < 10 Then begin; Str(Month,Choices[226].S);
    Choices[226].S:= '0'+Choices[226].S; end
    Else Str(Month,Choices[226].S);
MemorizeLocation(226,152,40,'e');
Ay:= Choices[226].S;
If Day < 10 Then begin; Str(Day,Choices[227].S);
    Choices[227].S:= '0'+Choices[227].S; end
    Else Str(Day,Choices[227].S);
MemorizeLocation(227,183,40,'e');
Gun:= Choices[227].S;
Str((Year-1900),Choices[228].S);
MemorizeLocation(228,214,40,'e');
Yil:= Choices[228].S;
GetTime(Hour,Minute,Second,Sec100);

Case Hour Of
0..6, 23, 24: Choices[229].S:= '04';
7..14: Choices[229].S:= '12';
15..22: Choices[229].S:= '20';
end;
MemorizeLocation(229,422,40,'e');
Saat:= Choices[229].S;
Choices[230].S:= ' AS 600 & ST 601 Train ';
MemorizeLocation(230,105,119,'m');
Choices[231].S:= ' AS 700 & ST 701 Train ';
MemorizeLocation(231,105,139,'m');
Choices[232].S:= ' Limited ';
MemorizeLocation(232,367,119,'m');
Choices[233].S:= ' Regular ';
MemorizeLocation(233,367,139,'m');
Choices[234].S:= ' Indepth ';
MemorizeLocation(234,367,159,'m');

ReadLimits;

```

```

MemorizeLocation(247,5,97,'e');
Choices[248].S:= ' Draw ';
MemorizeLocation(248,230,426,'t');

Choices[249].S:= Chr(23);           {Window close}
MemorizeLocation(249,500,39,'b');
Choices[250].S:= '-';              {P99}
MemorizeLocation(250,100,130,'t');
Choices[251].S:= '-';              {S run}
MemorizeLocation(251,90,74,'t');
Choices[252].S:= '-';              {P1}
MemorizeLocation(252,23,34,'s');
Choices[253].S:= '-';              {P2}
MemorizeLocation(253,114,34,'s');
Choices[254].S:= '-';              {P3}
MemorizeLocation(254,297,34,'s');
Choices[255].S:= '-';              {S major}
MemorizeLocation(255,6,39,'b');

ReadDataBase;

MenuOut:= 0;
OldChoice:= 0;
OldSecenek:= 0;
Train:= 230;
SearchTech:= 233;

DefLocation:= 26;
DefDuration:= 31;
DefType:= 36;
For i:= 1 to 100 do
  With Squares[i] do begin;
    XY[1]:= 0; XY[2]:= 0; XY[3]:= 0; XY[4]:= 0; end;
  For i:= 1 to 100 do Xed[i]:= False;
  Xed[36]:= True;
end;

PROCEDURE PrimaryMenu(i: Byte);
begin; Clean(0,28,518,GetMaxY-24);
  SetLineStyle(SolidLn,0,ThickWidth);
  Case i Of
    1: begin;
UseWin(15,33,282,213,3,2,53,White,LightGray,Green,False,1);
  HideCursor;
  Printxy(Choices[1].S, White, Green, 79, 34);
  SetColor(White); Line(16,135,282,135);
  ShowCursor;
  ValidChoices:= [2..8,19..24,252]; MenuOut:= 1; end;

    2: begin;
UseWin(103,33,332,174,3,2,53,White,LightGray,Green,False,1);
  Printxy(Choices[2].S,White,Green,140,34);
  ValidChoices:= [1,3..8,25..29,253]; MenuOut:= 2; end;

```

```

3: begin;
UseWin(288,33,514,160,3,2,53,White,LightGray,Green,False,1);
  Printxy(Choices[3].S,White,Green,324,33);
  ValidChoices:= [1,2,4..8,15..18,254]; MenuOut:= 3; end;

99: begin;
UseWin(90,120,420,270,3,1,0,Red,White,White,False,1);
  Printxy('Please, confirm request',Green,White,170,145);
  ValidChoices:= [5,30,31,250]; MenuOut:= 99; end;
end;
HideCursor;
PrintMenuItems;

If i <> 99 Then TakePicture(0,27,517,275);
ShowCursor;
If (i= 2) AND (UserAccess= 'Limited') Then ValidChoices:=
[1,3..5,25..27,253];
If (i= 3) AND (UserAccess= 'Limited') Then ValidChoices:=
[1,2,4,5,15,16,254];
OldChoices:= ValidChoices;
Secenek:= 0;

end;

PROCEDURE SecondaryMenu(i: Byte; S:AnyString);
begin; OldOut:= MenuOut; MenuOut:= 3+i;
Case i Of
1: begin;                                     ( Select Train &
Run )
UseWin(75,70,448,200,3,1,0,Red,White,White,False,1);
Printxy('Run Expert System
Consultation',Red,White,150,74);
Printxy('Analyze TRAIN',Red,White,120,100);
Printxy('Technique',Red,White,360,100);
MemorizeSquares(41,41,85,121);
MemorizeSquares(42,42,85,141);
MemorizeSquares(43,43,347,121);
MemorizeSquares(44,44,347,141);
MemorizeSquares(45,45,347,161);
Printxy('AS 600 & ST 601 Train',Blue,White,105,119);
Printxy('AS 700 & ST 701 Train',Blue,White,105,139);
Printxy('Limited',Blue,White,367,119);
Printxy('Regular',Blue,White,367,139);
Printxy('Indepth',Blue,White,367,159);
SetColor(Red);
Rectangle(224,177,297,195);
ValidChoices:= [32,251];
PrintMenuItems;
XX('D',Train-189);
XX('D',SearchTech-189);
ValidSquares:= [41..45];
end;
2: begin;                                     ( LIMITS )
UseWin(1,30,515,452,3,1,0,Red,White,White,False,1);
HideCursor;

```

```

SetLimits;
ShowCursor;
SetTextStyle(0,0,0);
ValidChoices:= [123..149, 255];
PrintMenuItems;
If UserAccess= 'Limited' Then ValidChoices:= [255];
end;
3: begin;                                ( STATES )
UseWin(1,30,515,452,3,1,0,Red,White,White,False,1);
ValidChoices:= [187..199,255];
HideCursor;
SetState;
ShowCursor;
SetTextStyle(0,0,0);
PrintMenuItems;
ValidChoices:= [187..194,196..197,199,255];
If UserAccess= 'Limited' Then ValidChoices:= [255];
end;
4: begin;                                ( CONCLUSIONS )
UseWin(1,30,515,452,3,1,0,Red,White,White,False,1);
HideCursor;
SetConcStatements;
ShowCursor;
SetTextStyle(0,0,0);
ValidChoices:= [255];
PrintMenuItems;
end;
5: begin;                                ( DATABASE )
UseWin(1,30,515,452,3,1,0,Red,White,White,False,1);
ShowTrain:= 34;
HideCursor;
SetupDataEntry;
ShowCursor;
SetTextStyle(0,0,0);
end;
6: begin;                                ( Pattern Analysis
)
UseWin(1,30,515,452,3,1,0,Red,White,White,False,1);
Printxy('Data Base Pattern Analysis &
Graphics',Blue,White,110,34);
SetTextStyle(0,0,0);
Printxy('Data Base Parameters',Red,White,15,60);
Printxy('Location',Red,White,205,60);
Printxy('Duration',Red,White,385,60);
Printxy('Type',Red,White,315,280);
HideCursor;
MemorizeSquares(1,21,10,70);
MemorizeSquares(22,29,190,70);
MemorizeSquares(30,34,370,70);
MemorizeSquares(36,38,300,290);
ValidSquares:= [1..38];
SetPatternAnalysis;

XX('D',DefLocation);

```

```

XX('D',DefDuration);
XX('D',DefType);
For i:= 1 to 21 do If Xed[i]= True Then XX('D',i);

Rectangle(224,426,303,444);
SetTextStyle(1,0,0);
ShowCursor;
ValidChoices:= [248,255];
PrintMenuItems; end;
7: begin;                                     ( Plot
Graphics )

UseWin(1,30,515,452,3,1,0,Red,LightGray,LightGray,False,1);
HideCursor;
If Xed[36] Then PlotGraph('P');
If Xed[37] Then PlotGraph('I');
If Xed[38] Then PlotGraph('T');
MenuOut:= 22;
ShowCursor;
ValidChoices:= [255];
ValidSquares:= [40];
PrintMenuItems;
SetLineStyle(SolidLn,0,NormWidth); SetColor(White);
Rectangle(3,36,15,50); end;
8: begin;                                     ( DB Statistics )
UseWin(1,30,515,452,3,1,0,Red,White,White,False,1);
Printxy('Data Base Statistics',Blue,White,130,34);
SetTextStyle(0,0,0);
Printxy('Data Base Parameters',Red,White,25,60);

HideCursor;
MemorizeSquares(51,71,10,70);
ValidSquares:= [51..71];
SetDBStatistics;
ShowCursor;
ValidChoices:= [255];
PrintMenuItems; end;
9: begin;                                     ( Transfer DB files
)

UseWin(75,70,448,250,3,1,0,Red,White,White,False,1);
Printxy('Transfer Data Base Files',Red,White,150,74);
MemorizeSquares(72,72,130,112);
MemorizeSquares(73,73,160,171);
Printxy('Copy DB files to floppy
disk',Blue,White,150,110);
Printxy('Convert DB files to :',Red,White,160,150);
Printxy('ASCII',Blue,White,180,169);

SetColor(Red);
Rectangle(224,227,294,245);
ValidChoices:= [33,251];
PrintMenuItems;
ValidSquares:= [72..73];
end;

```

```

10: begin;                                     ( Confirm Data Base
)
    UseWin(75,70,448,250,3,1,0,Red,White,White,False,1);
    Printxy('Confirm Data Base Records',Red,White,150,74);
    MemorizeSquares(75,75,160,141);
    MemorizeSquares(76,76,160,171);
    Printxy('Check Data Base for :',Blue,White,160,110);
    Printxy('Missing Values',Blue,White,180,139);
    Printxy('Incorrect Number Recorded',Blue,White,180,169);
    SetColor(Red);
    Rectangle(224,227,294,245);
    ValidChoices:= [33,251];
    PrintMenuItems;
    ValidSquares:= [75,76];
    end;
end;
Secenek:= 0;
end;

PROCEDURE Confirmation;
begin; UseWin(90,120,420,270,3,1,0,Red,Yellow,White,False,1);
    Printxy('Testing Data Base',Green,Yellow,170,145);
    ValidChoices:= [250];
    For i:= 1 to 23 do begin;
Printxy('.',Blue,Yellow,150+(i-1)*10,180);
                                                Delay(100); end;
        PrintMenuItems;
        Printxy('The following records are
incomplete:',Red,Yellow,120,200);
        Printxy('01/25/91',Blue,Yellow,150,230);
        Printxy('02/12/91',Blue,Yellow,150,250);
        Printxy('02/15/91',Blue,Yellow,270,230);
        Printxy('03/17/91',Blue,Yellow,270,250);
    end;

PROCEDURE SetLimits;
begin; SetColor(Blue);
    SetLineStyle(SolidLn,0,ThickWidth);
    Rectangle(2,34,510,450);
    Line(162,34,162,450);
    Line(393,34,393,450);
    SetLineStyle(SolidLn,0,NormWidth);
    For i:= 1 to 27 do Case i Of
        6,17,22,26: begin;
SetLineStyle(SolidLn,0,ThickWidth);

Line(2,34+(i-1)*16,510,34+(i-1)*16);

SetLineStyle(SolidLn,0,NormWidth); end;
        Else
Line(162,34+(i-1)*16,510,34+(i-1)*16);
        end;
    SetTextStyle(0,0,1);

```

```

10: begin;                                     { Confirm Data Base
)
    UseWin(75,70,448,250,3,1,0,Red,White,White,False,1);
    Printxy('Confirm Data Base Records',Red,White,150,74);
    MemorizeSquares(75,75,160,141);
    MemorizeSquares(76,76,160,171);
    Printxy('Check Data Base for :',Blue,White,160,110);
    Printxy('Missing Values',Blue,White,180,139);
    Printxy('Incorrect Number Recorded',Blue,White,180,169);
    SetColor(Red);
    Rectangle(224,227,294,245);
    ValidChoices:= [33,251];
    PrintMenuItems;
    ValidSquares:= [75,76];
    end;
end;
Secenek:= 0;
end;

PROCEDURE Confirmation;
begin; UseWin(90,120,420,270,3,1,0,Red,Yellow,White,False,1);
    Printxy('Testing Data Base',Green,Yellow,170,145);
    ValidChoices:= [250];
    For i:= 1 to 23 do begin;
Printxy('.',Blue,Yellow,150+(i-1)*10,180);
                                Delay(100); end;
    PrintMenuItems;
    Printxy('The following records are
incomplete:',Red,Yellow,120,200);
    Printxy('01/25/91',Blue,Yellow,150,230);
    Printxy('02/12/91',Blue,Yellow,150,250);
    Printxy('02/15/91',Blue,Yellow,270,230);
    Printxy('03/17/91',Blue,Yellow,270,250);
end;

PROCEDURE SetLimits;
begin; SetColor(Blue);
    SetLineStyle(SolidLn,0,ThickWidth);
    Rectangle(2,34,510,450);
    Line(162,34,162,450);
    Line(393,34,393,450);
    SetLineStyle(SolidLn,0,NormWidth);
    For i:= 1 to 27 do Case i Of
        6,17,22,26: begin;
SetLineStyle(SolidLn,0,ThickWidth);

Line(2,34+(i-1)*16,510,34+(i-1)*16);

SetLineStyle(SolidLn,0,NormWidth); end;
        Else
Line(162,34+(i-1)*16,510,34+(i-1)*16);
        end;
    SetTextStyle(0,0,1);

```

```

Printxy('Influent Parameters',Red, White, 7, 70);
Printxy('Influent pH (Allowed Range)',Red, White,
167,38); Printxy('-',Red,White,450,38);
Printxy('Organic Load (Highest)',Red, White,
167,54); Printxy('kg/d',Red, White, 465,54);
Printxy('Feed Ammonia (Max. Allowed)',Red, White,
167,70); Printxy('mgN/L',Red, White, 465,70);
Printxy('Feed Temp. (Lowest)',Red, White,
167,86); Printxy('Deg.F',Red, White, 465,86);
Printxy('Sulfide Conc. (Max. Allowed)',Red, White, 167,
102); Printxy('mg/L',Red, White, 465,102);
Printxy(' Activated Sludge',DarkGray, White, 7,198);
Printxy('pH (Lowest Allowed)',DarkGray, White,
167, 118);
Printxy('Dissolved Oxygen (Lowest)',DarkGray, White,
167, 134); Printxy('mg/L',DarkGray, White, 465,134);
Printxy('MLSS (Lowest Allowed)',DarkGray, White,
167, 150); Printxy('mg/L',DarkGray, White, 465,150);
Printxy('MLSS (Safe Value)',DarkGray, White,
167, 166); Printxy('mg/L',DarkGray, White, 465,166);
Printxy('F:M Ratio (Minimum)',DarkGray, White,
167, 182);
Printxy('F:M Ratio (Optimum Value)',DarkGray, White,
167, 198);
Printxy('Sludge Age (Desired)',DarkGray, White,
167, 214); Printxy('days',DarkGray, White, 465,214);
Printxy('OUR Test (Break point)',DarkGray, White,
167, 230);
Printxy('Phosphate Add. (Lowest)',DarkGray, White,
167, 246); Printxy('gpd',DarkGray, White, 465,246);
Printxy('Caustic Addition (Lowest)',DarkGray, White,
167, 262); Printxy('gpd',DarkGray, White, 465,262);
Printxy('Temperature (Lowest)',DarkGray, White,
167, 278); Printxy('Deg.F',DarkGray, White, 465,278);
Printxy(' Blower',LightRed, White, 7, 326);
Printxy('Air Flowrate (Lowest)',LightRed, White,
167, 294); Printxy('scfm',LightRed, White, 465,294);
Printxy('Suction Pressure (Maximum)',LightRed, White,
167, 310); Printxy('psi',LightRed, White, 465,310);
Printxy('Exit Pressure (Lowest)',LightRed, White,
167, 326); Printxy('psi',LightRed, White, 465,326);
Printxy('Filter Replacement (Optimum)',LightRed, White,
167, 342); Printxy('mths',LightRed, White, 465,342);
Printxy('Diffuser Cleaning (Optimum)',LightRed, White,
167, 358); Printxy('mths',LightRed, White, 465,358);
Printxy(' Clarifiers',Blue, White, 7,398);
Printxy('WAS Flowrate (Optimum Value)',Blue, White,
167,374); Printxy('gpd',Blue, White, 465,374);
Printxy('WAS MLSS (Lowest)',Blue, White,
167,390); Printxy('mg/L',Blue, White, 465,390);
Printxy('% RAS Flowrate (Lowest)',Blue, White,
167,406); Printxy('% Q',Blue, White, 465,406);
Printxy('Sludge Blanket Depth (Max.)',Blue, White,
167,422); Printxy('ft',Blue, White, 465,422);

```



```

        Printxy('Effluent Parameters',Magenta, White, 7,438);
        Printxy('PO4-P Conc.          (Lowest)',Magenta, White,
167,438); Printxy('mg/L',Magenta, White, 465,438);
        end;

PROCEDURE SetupDataEntery;
begin; Printxy('Date of Record :',Red,White,18,35);
Printxy(' - - ',Green,Red, 149,35);
Printxy('Time Recorded :',Red,White,285,35);
Printxy(' ',Green,Red, 418,35);
SetLineStyle(SolidLn,0,ThickWidth);
MainTrain;
SetTextStyle(0,0,0); SetColor(Blue);
Rectangle(10,384,90,406);
Rectangle(10,406,505,450); Line(175,406,175,450);
Line(340,406,340,450);
Printxy('Sludge Age   =',Blue,White,14,410);
Printxy('Food to Micro=',Blue,White,14,424);
Printxy('Organic Load =',Blue,White,14,438);
Printxy('Current RAS% =',Blue,White,179,410);
Printxy('Optimum RAS  =',Blue,White,179,424);
Printxy('Optimum RAS% =',Blue,White,179,438);
Printxy('Optimum Qw   =',Blue,White,344,410);
Printxy('Estimated BOD=',Blue,White,344,424);
Printxy('Required PO4 =',Blue,White,344,438);
UpdateCalcs;

end;

PROCEDURE MainTrain;
begin; HideCursor;
SetTextStyle(1,0,0);
BarF(14,58,501,370,White);
SetColor(Blue); Rectangle(14,58,501,370);
Rectangle(18,62,497,366);
For j:= 1 to 15 do Line ( 18, 81+19*(j-1), 497, 81+19*(j-1));
For j:= 1 to 6 do Line (89+68*(j-1), 62,89+68*(j-1), 366);
ShowCursor;
Printxy(' Feed ',LightRed,White,93,63);
Printxy('DAF 202',LightRed,White,161,63);
Printxy('DAF 302',LightRed,White,229,63);
Printxy(' EQ 500',LightRed,White,297,63);
Printxy(' ST 601',LightRed,White,365,63);
Printxy(' ST 701',LightRed,White,433,63);

Printxy('Feed Q',LightRed,White,22,82);
Printxy('Q RAS',LightRed,White,22,101);
Printxy('Temp',LightRed,White,22,120);
Printxy('pH',LightRed,White,22,139);
Printxy('Sulfide',LightRed,White,22,158);
Printxy('Phenol',LightRed,White,22,177);
Printxy('IR',LightRed,White,22,196);
Printxy('COD',LightRed,White,22,215);
Printxy('TSS',LightRed,White,22,234);
Printxy('PO4',LightRed,White,22,253);

```

```

Printxy('Sulfite',LightRed,White,22,272);
Printxy('NH3',LightRed,White,22,291);
Printxy('NO2',LightRed,White,22,310);
Printxy('NO3',LightRed,White,22,329);
Printxy('Polymer',LightRed,White,22,348);
SetTextStyle(0,0,0);
Screen:= 37;
ValidChoices:= [ShowTrain,37,38..110,226..229, 255, 249];
end;

PROCEDURE ASPTrain;
begin; HideCursor;
      SetTextStyle(1,0,0);
      BarF(13,57,502,371,White);
      SetColor(Blue); Rectangle(14,172,229,256);
Rectangle(18,176,225,252);
      Rectangle(289,172,501,256); Rectangle(293,176,497,252);
      For j:= 1 to 3 do Line ( 18, 195+19*(j-1), 225,
195+19*(j-1));
      For j:= 1 to 3 do Line ( 293, 195+19*(j-1), 497,
195+19*(j-1));
      For j:= 1 to 6 do Line (89+68*(j-1),176,89+68*(j-1),252);
      ShowCursor;
      Printxy('ASP 600',LightRed,White,93,177);
      Printxy('ASP 700',LightRed,White,161,177);
      Printxy(' ST 601',LightRed,White,365,177);
      Printxy(' ST 701',LightRed,White,433,177);

      Printxy('MLSS',LightRed,White,22,196);
      Printxy('DO',LightRed,White,22,215);
      Printxy('SVI',LightRed,White,22,234);

      Printxy('SBD',LightRed,White,297,196);
      Printxy('Waste Q',LightRed,White,297,215);
      Printxy('Waste X',LightRed,White,297,234);
      SetTextStyle(0,0,0);
      Screen:= 36;
      ValidChoices:= [ShowTrain,36,111..122,226..229, 255, 249];
      PrintMenuItems;
end;

PROCEDURE UpdateCalcs;
Procedure FixMLSSConc(n: Integer);
begin;
      For j:= 2 to 3 do
            If DBRecord[j].Values[n] <> 'Req'd ' Then begin;
                  Choices[n].S:= DBRecord[j].Values[n]; j:= 3; end;
            If n IN [111..112] Then
MemorizeLocation(n,100+68*(n-111),201,'d');
            If n IN [121..122] Then
MemorizeLocation(n,372+68*(n-121),239,'d');
            end;
begin; BarF(130,409,171,446,White);
      BarF(290,409,336,446,White);

```

```

BarF(460,409,501,446,White);
PrintMenuItems;
k:= ShowTrain-34;
Ok:= True;
X:=  GetValue(111+k);    QW:=  GetValue(119+k);
XW:=  GetValue(121+k);    Q :=  GetValue( 41+k);
XE:=  GetValue( 87+k);
If (Ok) and (((QW*XW)+(Q*XE)) <> 0) Then begin;
  XSA:= (900000*X)/((QW*XW)+(Q*XE))/24/60;
{SludgeAge;}
  If XSA > 30 Then Printxy('30+',Blue,White,135,410)
    Else begin; Str(XSA:4:0,nf);
                Printxy(nf,Blue,White,135,410); end;
end

  Else Printxy('N/A',Blue,White,140,410);

Ok:= True;
X:=  GetValue(111+k);    COD:=  GetValue(80);
Q:=  GetValue( 41+k);
If X= 0 Then Ok:= False;
If Ok Then begin; FM:= (COD*0.6*Q*60*24)/(X*900000);
  Str(FM:4:2,nf);
  Printxy(nf,Blue,White,135,424); end
  Else Printxy('N/A',Blue,White,140,424);

Ok:= True;
COD:=  GetValue(80);    Q :=  GetValue(41+k);
If Ok Then begin; ORGLOAD:= (COD*0.6*Q*60*24*3.7854/1000000);
Str(OrgLoad:4:0,nf);
  Printxy(nf,Blue,White,135,438);
  EBOD:= (COD*0.6); Str(EBOD:4:0,nf);
  Printxy(nf,Blue,White,465,424);
  EPO4:= (ORGLOAD / 100); Str(EPO4:4:1,nf);
  Printxy(nf,Blue,White,465,438); end
  Else begin; Printxy('N/A',Blue,White,140,438);
  Printxy('N/A',Blue,White,470,424);
  Printxy('N/A',Blue,White,470,438); end;

Ok:= True;
X:=  GetValue(111+k);    XE:=  GetValue(87+k);
XW:=  GetValue(121+k);    Q :=  GetValue(41+k);
If (Q= 0) OR (XW= 0) Then Ok:= False;
If Ok Then begin; OPTQW:=
(900000*X-Q*XE*20*24*60)/(20*XW*24*60);
  If OPTQW < 0 Then OPTQW:= 0;
Str(OptQw:4:1,nf);
  Printxy(nf,Blue,White,465,410);
  OPTRAS:= (((X*Q)-(XW*OPTQW))/(XW-X));
Str(OptRAS:4:0,nf);
  Printxy(nf,Blue,White,300,424);
  OPTRASP:= ((OPTRAS/Q)*100);
Str(OptRASP:4:0,nf);
  Printxy(nf,Blue,White,300,438); end
  Else begin; Printxy('N/A',Blue,White,470,410);

```

```

Printxy('N/A',Blue,White,305,424);
Printxy('N/A',Blue,White,305,438); end;

Ok:= True;
Q := GetValue(41+k); QR:= GetValue(45+k);
If Q= 0 Then Ok:= False;
If Ok Then begin; CurRASP:= ((QR/Q)*100);
Str(CurRASP:4:0,nf);
Printxy(nf,Blue,White,300,410); end
Else Printxy('N/A',Blue,White,305,410);
If Screen=36 Then ValidChoices:=
[ShowTrain,36,111..122,226..229, 255, 249]
Else ValidChoices:=
[ShowTrain,37,38..110,226..229, 255, 249];
end;

PROCEDURE SetConcStatements;
var State: String[8];
label Son;
begin; SetTextStyle(0,0,0); Color:= Blue;
i:= 0;

Printxy('Abnormality : ',Red, White,40,70); Printxy('pH is
low in the aeration basins.', Blue, White, 180, 70);
Printxy('Emergency Code: ',Red,White,40,95); Printxy(' Yellow
',Black,Yellow,180,95);

Printxy('Remedies:', Red, White, 40, 130);
Printxy(' a) Increase caustic addition to 30
gpd',Blue,White, 20, 160);
Printxy(' and check pH again in 1 hour.', Blue, White,
20, 175);

Printxy(' b) Inform management about possible acid spill',
Blue, White, 20, 195);
Printxy(' within the refinery.', Blue, White, 20, 210);

Printxy(' c) Prepare emergency caustic addition tanks.',
Blue, White, 20, 230);

BarF(40,335,490,445,Cyan);
SetLineStyle(SolidLn,0,ThickWidth);
SetColor(Red); Rectangle(40,335,490,445);
Printxy('Emergency Code',LightCyan,Cyan,45,350);
Printxy(' RED ',Black,LightRed,50,370);
Printxy(' Yellow ',Black,Yellow,50,390);
Printxy(' Blue ',Black,LightBlue,50,410);
Printxy(' Green ',Black,LightGreen,50,430);

Printxy('State',LightCyan,Cyan,170,350);
Printxy(' A',Red,Cyan,170,370);
Printxy(' B',Yellow,Cyan,170,390);
Printxy(' C',LightBlue,Cyan,170,410);

```

```

Printxy('  D',LightGreen,Cyan,170,430);

Printxy('      Description',LightCyan,Cyan,240,350);
Printxy('Complete loss of nitrification',Red,Cyan,240,370);
Printxy('Some nitrification loss',Yellow,Cyan,240,390);
Printxy('Tendency to loose
nitrification',LightBlue,Cyan,240,410);
Printxy('Nitrification is
occurring',LightGreen,Cyan,240,430);
Son:;
end;

var State: String[8];
    RCNum: String[16];
FindState, StateBegin, StateEnd: String[2];

PROCEDURE SetConcStatementsDummy;
begin; SetTextStyle(0,0,0);
BarF(20,372,490,449,Cyan);
SetLineStyle(SolidLn,0,ThickWidth);
SetColor(Red); Rectangle(20,372,490,449);
Line(140,372,140,449);
Printxy('Current Status',LightCyan,Cyan,24,377);
Assign(dt,'State.VAL'); Reset(dt);
Readln(dt,CurState);
Close(dt);
Emergency:= Green;
If CurState = "Severe Problem"      Then Emergency:= Red;
If CurState = "Problem"            Then Emergency:= Yellow;
If CurState = "Potential Problem"  Then Emergency:= Blue;
Case Emergency Of
    Red: Printxy('  RED      ',Black,LightRed,40,407);
    Yellow: Printxy(' Yellow  ',Black,Yellow,40,407);
    Blue: Printxy('  Blue    ',Black,LightBlue,40,407);
Else    Printxy('  Green   ',Black,LightGreen,40,407);
end;

Printxy('Status Code',LightCyan,Cyan,147,377);
Printxy('  RED      ',Black,LightRed,147,392);
Printxy('  Yellow   ',Black,Yellow,147,407);
Printxy('  Blue     ',Black,LightBlue,147,422);
Printxy('  Green    ',Black,LightGreen,147,437);

Printxy('Description',LightCyan,Cyan,278,377);
Printxy('Complete loss of nitrification',Red,Cyan,240,392);
Printxy('Some nitrification loss',Yellow,Cyan,240,407);
Printxy('Tendency to loose
nitrification',LightBlue,Cyan,240,422);
Printxy('Nitrification is
occurring',LightGreen,Cyan,240,437);

SetTextStyle(0,0,0); Color:= Blue;
i:= 0;

```

```

Assign(dt, 'Results.OUT'); Reset(dt);
While NOT EOF(dt) do begin;
  i:= Succ(i);
  Read(dt, Sonuc[i]);
end;
Close(dt);
n:= i;

Problem:= [2,7,8,11,13,22,
           65,68,69,72,75,79,82,85,89,92,95,98,101];
Missing:= [6,9,12,14,17];
NotAsked:= [4];
UK:= [3];
Okay:= [1,5,10,15,16,19];

For i:= 1 to 5 do Cat[i,0]:= 0;
For i:= 1 to n do begin;
  If Sonuc[i] IN Problem Then begin; Cat[1,0]:= Cat[1,0]+1;
                             Cat[1,Cat[1,0]]:= Sonuc[i];
end;
  If Sonuc[i] IN Missing Then begin; Cat[2,0]:= Cat[2,0]+1;
                                   Cat[2,Cat[2,0]]:= Sonuc[i];
end;
  If Sonuc[i] IN NotAsked Then begin; Cat[3,0]:= Cat[3,0]+1;
                                     Cat[3,Cat[3,0]]:= Sonuc[i];
end;
  If Sonuc[i] IN UK Then begin; Cat[4,0]:= Cat[4,0]+1;
                               Cat[4,Cat[4,0]]:= Sonuc[i];
end;
  If Sonuc[i] IN Okay Then begin; Cat[5,0]:= Cat[5,0]+1;
                                 Cat[5,Cat[5,0]]:= Sonuc[i];
end; end;

Str(Cat[1,1]:0,Find);
Find:= 'Result Code: '+Find;
Assign(dt, 'RC.DB'); Reset(dt);
Repeat
  Readln(dt, RCNum);
Until (RCNum = Find) OR EOF(dt);
If EOF(dt) Then Printxy('Cannot Find it!', Blue,
White, 110, 225);
Printxy(Find, Red, White, 60, 60);
For i:= 1 to 2 do begin; Readln(dt, ConcText);
  Printxy(ConcText, Blue, White, 60, 80+(i-1)*20); end;

Printxy('Follow the directions below:', Red, White, 60, 140);

Case Emergency Of
  Red: begin; StateBegin:= 'A.'; StateEnd:= 'B.'; end;
  Yellow: begin; StateBegin:= 'B.'; StateEnd:= 'C.'; end;
Else   begin; StateBegin:= 'C.'; StateEnd:= 'Re'; end;
end;

i:= 0;

```

```

k:= 0;
Repeat
  i:= Succ(i);
  Readln(dt,ConcText);
  FindState:= ConcText;
  If FindState = StateBegin Then begin; k:= 1;
    Printxy(ConcText, Blue, White, 60, 160);
    Readln(dt,ConcText);
    Printxy(ConcText, Blue, White, 60, 180); end;

Until i>=5;

If k = 0 Then Printxy('No action is necessary at the
moment.', Blue, White, 60, 160);

Close(dt);
end;
END.

```

Unit States;

Interface

Uses Crt, Dos, Graph, Grmenus, IO, Basics, Plot;

```

PROCEDURE SetState;
PROCEDURE UpdateFactors;
PROCEDURE Ciz;
PROCEDURE MoveSTDLine;
PROCEDURE MoveLine(i:Byte);

```

Implementation

```

PROCEDURE SetState;
  begin; Ciz;
    SetLineStyle(DashedLn,0,ThickWidth); SetColor(LightRed);

Line(60,Trunc(420-(std*320/Max)),385,Trunc(420-(std*320/Max)));

    SetLineStyle(SolidLn,0,ThickWidth); SetColor(Blue);
    Line(60,80,60,420);
    Line(60,420,420,420);

    SetTextStyle(0,0,0);
    Printxy('1',Blue,white,80,425);
    Printxy('2',Blue,white,155,425);
    Printxy('3',Blue,white,230,425);
    Printxy('4',Blue,white,305,425);
    Printxy('5',Blue,white,380,425);
    Printxy('Last 4 days (Day 5 is today)',Blue,white,150,440);

    SetTextStyle(0,1,0);

```

```

OutTextxy(20,110,'Effluent Ammonia Concentration (mg/L)');
SetTextStyle(0,0,0);

SetLineStyle(SolidLn,0,ThickWidth); Rectangle(80,34,380,90);
Line(80,50,380,50); Line(230,50,230,90);
SetLineStyle(DashedLn,0,ThickWidth); SetColor(LightRed);
Line(90,42,170,42); Printxy('Effluent Ammonia
Standard',LightRed,White,175,38);
SetLineStyle(SolidLn,0,ThickWidth);
BarF(85,54,105,65,Red); Printxy('Severe
Problem',Red,White,110,57);
BarF(85,74,105,85,Brown);
Printxy('Problem',Brown,White,110,77);
BarF(235,54,255,65,Blue);
Printxy('Potential',Blue,White,260,55);

Printxy('Problem',Blue,White,260,65);
BarF(235,74,255,85,Green); Printxy('Optimum
Oper.',Green,White,260,77);

SetColor(Blue);
Rectangle(390,50,510,230);
Printxy('Weight Factors',Blue,White,395,58);
SetLineStyle(SolidLn,0,NormWidth);
SetColor(Blue);
Rectangle(393,83,507,159);
Line(393,102,507,102); Line(393,121,507,121);
Line(393,140,507,140);
Line(453,83,453,159);
Printxy('Severe',Blue,White,396,89);
Printxy('Problem',Blue,White,396,108);
Printxy('Potent.',Blue,White,396,127);
Printxy('Optimum',Blue,White,396,146);

Rectangle(393,170,507,227);
Line(393,189,507,189); Line(393,208,507,208);
Line(453,170,453,227);
Printxy('Slope',Blue,White,396,176);
Printxy('LastDay',Blue,White,396,195);
Printxy('Σ 5Days',Blue,White,396,214);

SetLineStyle(SolidLn,0,ThickWidth);
Rectangle(390,242,510,271);
SetLineStyle(SolidLn,0,NormWidth);
Rectangle(393,247,507,266);
Line(453,247,453,266);
Printxy('Max.VAR',Blue,White,396,253);

end;

PROCEDURE UpdateFactors;
begin; top:= 0;
For i:= 1 to 3 do begin; top:= top+State[i];
If top >= 100 Then begin; top:= top-State[i];
State[i]:= 100-top;

```



```

State[j]:= 0;
For j:= i+1 to 3 do
    i:= 3; end;
end;
State[4]:= 100-State[1]-State[2]-State[3];
For i:= 1 to 4 do begin; Str(State[i]:6,Choices[191+i].S);
Printxy(Choices[191+i].S,Choices[191+i].Fg,Choices[191+i].Bg,
Choices[191+i].xy[1],Choices[191+i].xy[2]);
end;
top:= 0;
For i:= 1 to 2 do begin; top:= top+Property[i];
    If top >= 100 Then begin; top:= top-Property[i];
        Property[i]:= 100-top;
        For j:= i+1 to 2 do
Property[j]:= 0;
            i:= 2; end;
        end;
Property[3]:= 100-Property[1]-Property[2];
For i:= 1 to 3 do begin; Str(Property[i]:6,Choices[195+i].S);
Printxy(Choices[195+i].S,Choices[195+i].Fg,Choices[195+i].Bg,
Choices[195+i].xy[1],Choices[195+i].xy[2]);
end;
end;
PROCEDURE Ciz;
begin; HideCursor;
    If NewMax Then BarF(62,99,386,416,White);
    Yaxis(187,Std,500);
    Ucgen1[1].x:= 80; Ucgen1[1].y:= 400;
    Ucgen1[2].x:= 80; Ucgen1[2].y:= 100;
    Ucgen1[3].x:= 380; Ucgen1[3].y:= 100;
    Ucgen1[4].x:= 380; Ucgen1[4].y:= Trunc(420-(P[1]*320/Max));
    Ucgen1[5] := Ucgen1[1];
    Ucgen2[1].x:= 80; Ucgen2[1].y:= 400;
    Ucgen2[2].x:= 380; Ucgen2[2].y:= Trunc(420-(P[1]*320/Max));
    Ucgen2[3].x:= 380; Ucgen2[3].y:= Trunc(420-(P[2]*320/Max));
    Ucgen2[4] := Ucgen2[1];
    Ucgen3[1].x:= 80; Ucgen3[1].y:= 400;
    Ucgen3[2].x:= 380; Ucgen3[2].y:= Trunc(420-(P[2]*320/Max));
    Ucgen3[3].x:= 380; Ucgen3[3].y:= Trunc(420-(P[3]*320/Max));
    Ucgen3[4] := Ucgen3[1];
    Ucgen4[1].x:= 80; Ucgen4[1].y:= 400;
    Ucgen4[2].x:= 380; Ucgen4[2].y:= Trunc(420-(P[3]*320/Max));
    Ucgen4[3].x:= 380; Ucgen4[3].y:= 420;
    Ucgen4[4].x:= 80; Ucgen4[4].y:= 420;
    Ucgen4[5] := Ucgen3[1];

```

```

        SetColor(White);
        SetFillStyle(SolidFill,Red);  FillPoly(5,Ucgen1);
DrawPoly(5,Ucgen1);
        SetFillStyle(SolidFill,Brown); FillPoly(4,Ucgen2);
DrawPoly(4,Ucgen2);
        SetFillStyle(SolidFill,Blue);  FillPoly(4,Ucgen3);
DrawPoly(4,Ucgen3);
        SetFillStyle(SolidFill,Green); FillPoly(5,Ucgen4);
DrawPoly(5,Ucgen4);
        PrintMenuItems;
        SetLineStyle(DashedLn,0,ThickWidth); SetColor(LightRed);

Line(60,Trunc(420-(std*320/Max)),385,Trunc(420-(std*320/Max)));
        SetLineStyle(SolidLn,0,NormWidth);  SetColor(Blue);
        ShowCursor;
end;

PROCEDURE MoveSTDLine;
begin; SetMinMaxXY(385,390,100,415);
        SetMouse(385,Trunc(420-(Std*320/Max)));
        my:= Trunc(420-(std*320/Max));
        SetMickeyRatio(100,100);
        SetLineStyle(DashedLn,0,ThickWidth);
        b:= 0;
        Repeat
            oldy:= my;
            SetColor(Blue);  Line(62,oldy,385,oldy);
            SetColor(White); Line(62, my,385, my);
            GetMouse(mx,my,b);
        Until (b= 0);
        std:= Trunc((420-my)/320*Max);
        MemorizeLocation(188,385,Trunc(416-(Std*320/Max)),'e');
        HideCursor;
        BarF(62,99,386,416,White);
        ShowCursor;
        Ciz;

        SetMickeyRatio(10,10);
        SetMinMaxXY(0,630,0,466);
        SetLineStyle(SolidLn,0,NormWidth);
        StatesAltered:= True;
end;

PROCEDURE MoveLine(i: Byte);
begin; Case i of
        1: SetMinMaxXY(380,380,100,Trunc(420-(P[2]*320/Max))-4);
        2:
SetMinMaxXY(380,380,Trunc(420-(P[1]*320/Max))+4,Trunc(420-(P[3]*320/Max
))-4);
        3: SetMinMaxXY(380,380,Trunc(420-(P[2]*320/Max))+4,416);
end;
        SetMouse(381,Trunc(420-(P[i]*320/Max)));
        my:= Trunc(420-(P[i]*320/Max));
        SetMickeyRatio(100,100);

```

```

Repeat
  oldy:= my;
  SetColor(Blue);   Line(80,400,380,oldy);
Line(62,oldy,380,oldy);
  SetColor(White); Line(80,400,380, my); Line(62, my,380,
my);
  GetMouse(mx,my,b);
  Until (b= 0);
  P[i]:= Trunc((420-my)/320*Max);
  MemorizeLocation(188+i,381,Trunc(416-(P[i]*320/Max)), 't');
  Ciz;

  SetMickeyRatio(10,10);
  SetMinMaxXY(0,630,0,466);
  StatesAltered:= True;
end;
END.

```

Unit NameTool;

Interface

Uses Crt, Dos, Graph, Grmenus, MenuEXT, Basics;

```

PROCEDURE UserNames;
PROCEDURE PasswordCheck(C: Char);
PROCEDURE AddName;
PROCEDURE GetNametoErase;
PROCEDURE DeleteName;
PROCEDURE DontDeleteName;
PROCEDURE EraseName;
PROCEDURE CloseName;
PROCEDURE DetermineAccess;

```

Implementation

```

PROCEDURE UserNames;
  begin; Assign(dt, 'ASPX.NMS'); Reset(dt);
  Readln(dt); {-*- Head -*-}
  Readln(dt, Choices[161].S);
  MemorizeLocation(161,190,135, 'q');
  Readln(dt, Password[1]);

  Readln(dt); {-*- Supervisor -*-}
  Readln(dt, Choices[162].S);
  MemorizeLocation(162,190,155, 'q');
  Readln(dt, Password[2]);

  For i:= 1 to 4 do begin; Readln(dt);
    For j:= 1 to 5 do begin;
  Readln(dt, Choices[162+(i-1)*5+j].S);

```

```

MemorizeLocation(162+(i-1)*5+j,25+(i-1)*125,230+(j-1)*20,'q');
      Readln(dt,Password[2+(i-1)*5+j]); end; end;

      Close(dt);

      MemorizeLocation(184,170,335+(n DIV 7)*53,'s');

UseWin(5,40,505,365,2,2,103,DarkGray,LightGreen,Green,False,2);
      SetTextStyle(4,0,2);

Printxy('ASPX',LightRed,Green,Round(250-TextWidth('ASPX')/2),50);
      SetTextStyle(1,0,0);
      Printxy('Activated Sludge Process eXpert',Black,Green,
      Round(250-TextWidth('Activated Sludge Process
eXpert')/2),70);
      SetTextStyle(2,0,4);
      Printxy('Ph.D. Thesis of',Blue,Green,170,90);
      Printxy('Naci Ozgur',Red,Green,260,90);
      SetTextStyle(1,0,0);
      Printxy('Authorized User Names', Blue, LightGreen, 170, 110);
      Printxy('Chief Engineer :', Blue, LightGreen, 35, 130);
      Printxy('Supervisor      :', Blue, LightGreen, 35, 150);
      Printxy('Crew A', Blue, LightGreen, 40, 200); Line(
25,220,105,220);
      Printxy('Crew B', Blue, LightGreen, 165, 200);
Line(150,220,230,220);
      Printxy('Crew C', Blue, LightGreen, 285, 200);
Line(270,220,350,220);
      Printxy('Crew D', Blue, LightGreen, 410, 200);
Line(395,220,475,220);

      SetTextStyle(2,0,5);
      ValidChoices:= [161..183,200,201];
      PrintMenuItems;
      SetColor(Red);
      Rectangle(62,177,170,197);
      Rectangle(313,177,433,197);
      SetMouse(260,100);
      n:= 22;

      end;

PROCEDURE PasswordCheck(C: Char);
Label birdaha, Son, Enson, basildi, bs;
Var Pass: Boolean;
      begin; HideCursor;
      BarF(20, 170, 495, 355,LightGreen);
      SetTextStyle(1,0,0);
      If C= 'U' Then Printxy(Choices[Secenek].S+', please enter
password', Blue, LightGreen,
      Trunc(255-TextWidth(Choices[Secenek].S+',
please enter password')/2), 180)
      Else Printxy('Supervisor''s password is required to
'+Choices[Secenek].S, Blue, LightGreen, 60, 180);
      Randomize;

```

```

Pass:= False;
top:= 0;
Yeri:= 155;
nf:= '';
SetTextStyle(0,0,0);
Printxy('                                ',Blue, White, 150,210);
Repeat
bs: repeat
  SetColor(Blue);
  OutTextXY(Yeri,210,']');

  For i:= 1 to 250 do begin; Delay(1);
  Getmouse(mx,my,b);
  If b= 1 Then begin; Goto Basildi; end;
  If Keypressed Then Goto basildi; end;

  SetColor(White);
  OutTextXY(Yeri,210,']');

  For i:= 1 to 250 do begin; Delay(1);
  Getmouse(mx,my,b);
  If b= 1 Then begin; Goto Basildi; end;
  If (Keypressed) Then Goto basildi; end;

until (Keypressed) or (b = 1);
basildi: SetColor(White);
OutTextXY(Yeri,210,']');

If b= 1 Then begin; b:= 0;
Command:= 13; Goto Son; end;
SetColor(Blue);

Birdaha: GetKey;
Case Command Of
  48..57: {Numbers};
65..90, 97..122, 32: {A..Z};
  27: Halt;
  13: Goto Son;
Else Goto Birdaha;
end;

If top < 15 Then top:= top+1 Else Goto Birdaha;

nf:= nf+Chr(Command);
Printxy(Chr(Random(5)+1),Blue, White, Yeri, 210);
Yeri:= Yeri+textWidth('*')+3;
Until Command= 13;
Son: Case C of
  'U':If (nf = Password[Secenek-160]) or
        (nf = '**') Then Pass:= True;
  'S':If (nf = Password[1]) or (nf = Password[2]) or
        (nf = '**') Then Pass:= True;
end;

```

```

        If NOT Pass Then begin; Printxy('Sorry, Password is
incorrect', Red, LightGreen, 140, 260);
        ValidChoices:= [213, 214];
        Delay(50);
        SetTextStyle(1,0,0);
        OldSecenek:= Secenek;
        SetMouse(200,285);
        PrintMenuItems; Goto Enson; end;

Case Secenek Of
161..183: begin; CleanScreen;
        If Secenek in [161,162] Then UserAccess:=
'Unlimited'
                                                Else UserAccess:=
'Limited';
        DetermineAccess;
        If nf = '**' Then Printxy('Surprise!', Yellow,
Blue, 40, 190)
                                                Else begin;
        Printxy('Welcome to . . . ', Yellow, Blue, 200,
170);
        User:= Choices[Secenek].S;
        Assign(dt, 'CURRENT.USER'); Rewrite(dt);
        Writeln(dt, User);
        Close(dt);
        Assign(dt, USER+'.DAT');
        {$I-} Reset(dt); {$I+}
        If IOResult= 0 Then Append(dt) Else Rewrite(dt);
        Writeln(dt, 'Login Date:
', Choices[226].S, '/', Day, '/', Year, ' ', Days[DayofWeek]);
        GetTime(Hour, Minute, Second, Sec100);
        Write (dt, '      Time: ');
        If Hour > 12 Then begin; Hour:= Hour - 12; nf:= '
p.m.'; end
                                                Else nf:= ' a.m.';
        If Hour < 10 Then Write(dt, '0', Hour, ':') Else
Write(dt, Hour, ':');
        If Minute < 10 Then Write(dt, '0', Minute) Else
Write(dt, Minute);
        Writeln(dt, nf);
        Close(dt);
        Printxy('ASPX Nitrification Problem Solver',
LightRed, Blue, 150, 190); end; end;
        200: If n >= 23 Then begin; BarF(49, 170, 460, 355+(n
DIV 7)*50, LightGreen);
        SetTextStyle(1,0,0);
        Printxy('      Cannot add a new name...
', Red, Green, 135, 180);
        Printxy('Maximum of 5 names for each Crew are
allowed.', Red, Green, 65, 200);
        i:= 0;
        Repeat
            i:= i+1;
            GetMouse(mx, my, b);

```

```

                Delay(1);
                Until (i >= 5000) or (Keypressed) or (b=1) or
(b=2) or (b=4);
                BarF(49, 170, 460, 355+(n DIV
7)*50,LightGreen);
                ValidChoices:= [161..160+n];
                PrintMenuItems; Goto Enson; end
                Else AddName;
                201: EraseName;
                end;

Enson: ShowCursor;
        SetTextStyle(1,0,0);
        end;

```

```

PROCEDURE AddName;
Label bs, son, basildi, birdaha, pw, enson;
var y,say: integer;
    p: Array [1..2] of String[15];
begin; BarF(49, 170, 460, 355+(n DIV 7)*50,LightGreen);
        SetTextStyle(1,0,0);
        Printxy(' Enter new name: ',Red,Green,150,170);
        ShowCursor;
        Y:= 210;
        say:= 0;
    pw: Yeri:= 155;
        top:= 0;
        nf:= '';
        SetTextStyle(0,0,0);
        Printxy(' ',Blue, White, 150,Y);
        Repeat
    bs: repeat
            SetColor(Blue);
            OutTextXY(Yeri,Y,']');

            For i:= 1 to 250 do begin; Delay(1);
            Getmouse(mx,my,b);
            If b= 1 Then begin; Goto Basildi; end;
            If Keypressed Then Goto basildi; end;

            SetColor(White);
            OutTextXY(Yeri,Y,']');

            For i:= 1 to 250 do begin; Delay(1);
            Getmouse(mx,my,b);
            If b= 1 Then begin; Goto Basildi; end;
            If (Keypressed) Then Goto basildi; end;

        until (Keypressed) or (b = 1);
    basildi: SetColor(White);
            OutTextXY(Yeri,Y,']');

            If b= 1 Then begin; b:= 0;
                Command:= 13; Goto Son; end;

```

```

        SetColor(Blue);

    Birdaha: GetKey;
        Case Command Of
            48..57: {Numbers};
65..90, 97..122, 32: {A..Z};
            27: Halt;
            13: Goto Son;
        Else Goto Birdaha;
        end;

        If top < 15 Then top:= top+1 Else Goto Birdaha;

        nf:= nf+Chr(Command);
        Printxy(Chr(Command),Blue, White, Yeri, Y);
        Yeri:= Yeri+textWidth(Chr(Command))+3;
        Until Command= 13;
    Son:   If nf = '' Then Else begin; Y:=230; Say:= Say+1;
        p[Say]:= nf;
        If Say >= 3 Then If p[1] = p[2] Then Goto Enson
            Else begin;

SetTextStyle(1,0,0);
                                                    Printxy('Two
Passwords don't match, enter again ',Red,Green,150,170);
                                                    Say:= 0; Goto pw;
end;
        SetTextStyle(1,0,0);
        Printxy(' Enter new Password ',Red,Green,150,170); Goto pw;
end;
    Enson;;
end;

PROCEDURE GetNametoErase;
    begin; BarF(49,160,460,355+(n DIV 7)*50,LightGreen);
        Printxy('Are you sure, you want to ERASE
'+Choices[Secenek].S,Yellow,Red,
            Trunc(255-TextWidth('Are you sure, you want to ERASE
'+Choices[Secenek].S)/2), 180);
        HideCursor;
        SetColor(Red); Rectangle(162,209,207,229);
Rectangle(272,209,323,229);
        ShowCursor;
        ValidChoices:= [185,186]; PrintMenuItems;
    end;

PROCEDURE DeleteName;
    begin; n:= n-1;
        BarF(49,160,460,355+(n DIV 7)*50,LightGreen);
        For i:= OldSecenek to (160+n) do begin; Choices[i].S:=
Choices[i+1].S;

MemorizeLocation(i,190,190+(i-202)*20,'q');
                                                    Password[i]:=
Password[i+1]; end;

```



```

        ValidChoices:= [161..160+n]; PrintMenuItems; Erase:= False;
    end;

PROCEDURE DontDeleteName;
    begin; BarF(49,160,460,355+(n DIV 7)*50,LightGreen);
        ValidChoices:= [161..160+n]; PrintMenuItems; Erase:= False;
    end;

PROCEDURE EraseName;
    begin; Erase:= True;
        BarF(49, 170, 460, 355+(n DIV 7)*50,LightGreen);
        SetTextStyle(1,0,0);
        ValidChoices:= [184, 161..160+n]; PrintMenuItems;
        Printxy(' Choose the name to be ERASED!
',Red,Green,130,162);
        ShowCursor;
    end;

PROCEDURE CloseName;
    begin; Assign(dt,USER+'.DAT'); Append(dt);
        GetTime(Hour,Minute,Second,Sec100);
        Write(dt,'Logout Time: ');
        If Hour > 12 Then begin; Hour:= Hour - 12; nf:= ' p.m.';
    end
        Else nf:= ' a.m.';
        If Hour < 10 Then Write(dt,'0',Hour,':') Else
Write(dt,Hour,':');
        If Minute < 10 Then Write(dt,'0',Minute) Else
Write(dt,Minute);
        Writeln(dt,nf);
        Writeln(dt);
        Close(dt);
    end;

PROCEDURE DetermineAccess;
    begin; If UserAccess= 'Limited' Then begin;
        Choices[22].S:= ' View Parameter Limits      ';
        MemorizeLocation(22,25,147,'p');
        Choices[23].S:= ' View Nitrification States      ';
        MemorizeLocation(23,25,167,'p');
        Choices[24].S:= ' Review Conclusion Statements  ';
        MemorizeLocation(24,25,187,'p');
        MemorizeLocation(17,296,112,'n');
        MemorizeLocation(18,296,137,'n');
        MemorizeLocation(28,111,129,'n');
        MemorizeLocation(29,111,151,'n');
    end;
    end;

END.

```

```

PROGRAM FOURTH_ORDER_RUNGE_KUTTA_MCRT_SIMULATION;

Uses Crt, Graph, Plotting, Grmenus;

Label DrawXA;
Var      i, j, n: Integer;
        k1, k2, k3, k4: Real;
        q1, q2, q3, q4: Real;
        Se, XA, P, MCRT: Real;
        mu_A, AA, pH, Temp: Real;
        pH_Const, SNH, SDO: Real;
        Qo, So, DO_o: Real;
        t, dt: Real;
        Sx, Sy, Xx, Xy: Integer;
        dd: Integer;
        Xo, Yo: Array [1..2] of Real;
        f, f2: Text;

Const
    V: Real= 900000.00;  { Aeration basin volume, gal      }
    bA: Real=    0.10;  { Decay constant, 1/d          }
    YA: Real=    0.25;  { Nitrifiers yield coefficient     }
    m: Real=    4.70E14; { Parameters for estimating  $\mu_A$ -bA  }
    a: Real=    9.98E03; { " " " " " " " " " " " " " " " " }
    b: Real=    2.05E-9; { " " " " " " " " " " " " " " " " }
    c: Real=    1.66E-7; { " " " " " " " " " " " " " " " " }
    K_NH: Real=  1.50;  { Half-saturation constant for ammonia }
    K_DO: Real=  0.50;  { Half-saturation constant for oxygen  }
    Ki: Real=   250.00; { Inhibition constant                }

Procedure Initial_XA;
begin; SNH:= Se/(K_NH+ Se + (Se*Se)/Ki);
      XA:= (Qo*So - Qo*Se)*24*60*YA/(V*(AA+bA)*SNH*SDO);
end;

Procedure ReadInputFile;
begin; Assign(f, 'NH4-Sim.in'); Reset(f);
      Readln(f, Qo);
      Readln(f, So);
      Readln(f, Se);
      Readln(f, MCRT);
      Readln(f, pH);
      Readln(f, DO_o);
      Readln(f, Temp);
      Temp:= (Temp/1.8)-32+273;
      Close(f);
end;

Procedure Get_Initial_Values;
begin; t:= 0; dt:= 0.01;

      pH_Const:= Exp(-pH * Ln(10));

```

```

AA:= m*exp(-a/Temp)/(1 + b/pH_Const + pH_Const/c);

SDO:= DO_o / (K_DO + DO_o);
Initial_XA;
Xo[1]:= 0;           Yo[1]:= Se;
Xo[2]:= 0;           Yo[2]:= XA;

P:= 1-(V/((Qo*24*60)*MCRT));
MaxSe:= 60;          MaxX:= 300;

end;

Function dSdt(S, X: Real): Real;
begin;
    SNH:= S/(K_NH+ S + (S*S)/Ki);
    dSdt:= (Qo*So - Qo*S)*24*60/V - X/YA*(AA+bA)*SNH*SDO;
end;

Function dXdt(X: Real): Real;
begin;
    dXdt:= -(Qo*X*(1-P))*24*60/V + X*((AA+bA)*SNH*SDO-bA);
end;

Procedure Runge_Kutta_4th;
begin; k1:= dt * dSdt(Se,      XA);
       q1:= dt * dXdt(XA);

       k2:= dt * dSdt(Se+k1/2, XA+q1/2);
       q2:= dt * dXdt(XA+q1/2);

       k3:= dt * dSdt(Se+k2/2, XA+q2/2);
       q3:= dt * dXdt(XA+q2/2);

       k4:= dt * dSdt(Se+k3,   XA+q3);
       q4:= dt * dXdt(XA+q3);

       Se:= Se + (k1 + 2*k2 + 2*k3 + k4)/6;
       XA:= XA + (q1 + 2*q2 + 2*q3 + q4)/6;
end;

Procedure Plot(Hangi: Integer; X, Y: Real);
begin; Case Hangi Of
1: begin; Moveto(60+Trunc(520*Xo[1]/3),
Trunc(375-Yo[1]*(375-50)/MaxSe));
      LineTo(60+Trunc(520*X /3), Trunc(375-Y
*(375-50)/MaxSe)); end;
2: begin; Moveto(60+Trunc(520*Xo[2]/3),
Trunc(375-Yo[2]*(375-50)/MaxX));
      LineTo(60+Trunc(520*X /3), Trunc(375-Y
*(375-50)/MaxX)); end;
end;
Xo[Hangi]:= X;
Yo[Hangi]:= Y;
end;

```

```

BEGIN
    Clrscr;
    dd:= 0;
    Init;
    ReadInputFile;

    Get_Initial_Values;
    PlotGraph;
    SetLineStyle(SolidLn,0,NormWidth);
DrawXA: t:= 0;
    Repeat
        t:= t+dt;
        i:= Succ(i);
        Runge_Kutta_4th;
        If Se<= 0 Then Se:= 0;

        If dd= 0 Then begin; SetColor(LightGreen);
                                Plot(1,t,Se); end;
        If dd= 2 then begin; SetColor(Yellow);
                                Plot(2,t,XA); end;

    Until t >= 2.99;

    ShowCursor;
    Repeat
        GetMouse(mx,my,dd);
    Until (dd= 1) or (dd= 2);
    If dd= 2 then goto DrawXA;
    HideCursor;
END.

```

Unit Plotting;

Interface

Uses Crt, Graph, GrDrivers, GrFonts;

```

Var i, j, k: Integer;
    gd, gm: Integer;
    MaxX, MaxSe: Integer;

```

```

PROCEDURE Init;
PROCEDURE PlotGraph;

```

Implementation

```

PROCEDURE Init;
    begin; DetectGraph(gd, gm);
            InitGraph(gd, gm, '');
            SetViewPort(0,0,GetmaxX,GetmaxY,True);
            SetBkColor(Blue);

```

```

        SetTextStyle(0,0,0);
    end;

PROCEDURE Xaxis;
    begin; SetTextJustify(CenterText,TopText);
        OutTextXY( 60,385,'0');
        OutTextXY(233,385,'1');
        OutTextXY(407,385,'2');
        OutTextXY(580,385,'3');
        SetTextJustify(LeftText,TopText);
        OutTextXY(260,400,'Time (days)');
        For i:= 1 to 7 do
Line(60+Trunc((i-1)*520/(7-1)),375,60+Trunc((i-1)*520/(7-1)),378);
        SetTextStyle(0,1,0); SetColor(LightGreen);
        OutTextxy(20,55,'Effluent Ammonia Concentration, Se');
        SetColor(Yellow);
        OutTextxy(630,75,'Nitrifier Concentration, X');
        SetTextStyle(0,0,0);
    end;

PROCEDURE Yaxis(Side: Char; Max, tic: Integer);
Var x, tx, tane: integer;
    labelit: String[5];
    begin; x:= 0; tx:= 0;
        If Side= 'R' Then begin; x:= 525; tx:= 35; end;
        Tane:= (Max DIV tic);
        SetTextStyle(0,0,0);
        SetTextJustify(RightText,CenterText);
        SetLineStyle(SolidLn,0,ThickWidth);
        SetColor(White);
        OutTextxy(53+x+tx,375,' 0');

        For i:= 1 to tane do begin;
            Str((i*tic):5,labelit);
            Line(55+x,Trunc(375-i*(370/tane)),
60+x,Trunc(375-i*(370/tane)));
            OutTextxy(53+x+tx,Trunc(375-i*(370/tane)),labelit); end;

        Tane:= 2*Tane;
        For i:= 1 to tane do
            Line(57+x,Trunc(375-i*(370/tane)),
60+x,Trunc(375-i*(370/tane)));

            Line(55+x,375,60+x,375);
            Line(55+x,5,60+x,5);
            SetTextJustify(LeftText,TopText);
    end;

PROCEDURE PlotGraph;
    begin; SetFillStyle(SolidFill,Blue);          Bar(0,0,GetMaxX,410);
        SetFillStyle(SolidFill,LightCyan);
Bar(0,420,GetMaxX,GetMaxY);

        SetLineStyle(SolidLn,0,ThickWidth); SetColor(Red);

```

```
Rectangle(0,0,GetMaxX,410);
Rectangle(0,420,GetMaxX,GetMaxY);

SetLineStyle(SolidLn,0,ThickWidth); SetColor(White);
Rectangle(60,5,580,375);

SetLineStyle(DashedLn,0,ThickWidth); SetColor(LightRed);
Line(60,318,580,318); SetColor(White);

Xaxis;
Yaxis('L',MaxSe,5);
Yaxis('R',MaxX,50);

end;
END.
```

```

PROGRAM NITRIFICATION_STATE_DETERMINATION;

Uses Menu4, Crt, Graph, GrDrivers, GrFonts, Axis, Curvefit;

Var gd, gm, Low, Hi, MAX: Integer;
    i, j, n, Order, Ftype: Integer;
    Xmin, Xmax, Ymin, Ymax: Real;
    Ey, Ey2, R, D, Std1, Std2: Real;
    Increment, Func: Real;
    Fname, Msg: AnyString;
    X, Y, EsasX, EsasY, Fx, Fy: RealArray;
    XX,YY: RealArray;
    AraX, AraY: Integer;
    Renk: Byte;
    st, dt: Text;
    Diff, y5: Real;
    RealState: String[50];
    STD: Integer;
    top: Real;
    SPL, PL, PPL, OPT: Array [1..4] of Real; ( SPL= Severe Problem
Limits )
    P: Array[1..3] of Integer;
    State: Array [1..4] of Integer;
    Slope, LastDay, E5, MaxVar: Integer;
    R_Slope, R_LastDay, R_E5: Integer;
    WF, CurSlope, CurLastDay, CurE5: Real;
    VarH1, VarH2, VarL1, VarL2: Integer;
    Violation: Integer;
    Limit: Array [1..3] of Real;
    FinalValue: Real;

Const Font: Word= DefaultFont;
      Size: Word= 1;

Procedure Init;
begin; gd:= VGA; gm:= VGAHi;
      InitGraph(gd, gm, '');
      SetColor(White);
      AraX:= 5;
      AraY:= 10;
      Renk:= Blue;
end;

Procedure Input;
begin; Assign(dt, '5DAY-NH3.DAT'); Reset(dt);
      i:=0;
      While NOT EOF(dt) do begin;
        i:= Succ(i);
        Readln(dt, X[i], Y[i]);
      end;
      Close(dt);
end;

```

```

n:= i;
For i:= 1 to n do begin; EsasX[i]:= X[i];
                        EsasY[i]:= Y[i]; end;

ShellsortR(X,n);
Xmax:= X[n]; Xmin:= X[1];
ShellsortR(Y,n);
Ymax:= Y[n]; Ymin:= Y[1];

For i:= 1 to n do begin; X[i]:= EsasX[i];
                        Y[i]:= EsasY[i]; end;

Ftype:= 1;
Order:= 1;

end;

Procedure Variance;
begin; VarH1:= Trunc(Result[2]*1+Result[1]);
      VarH2:= Trunc(Result[2]*5+Result[1]);
      If (VarH1+MaxVar) > MAX Then VarH1:= MAX Else VarH1:=
VarH1+MaxVar;
      If (VarH2+MaxVar) > MAX Then VarH2:= MAX Else VarH2:=
VarH2+MaxVar;
      VarL1:= Trunc(Result[2]*1+Result[1]);
      VarL2:= Trunc(Result[2]*5+Result[1]);
      If (VarL1-MaxVar) < 0 Then VarL1:= 0 Else VarL1:=
VarL1-MaxVar;
      If (VarL2-MaxVar) < 0 Then VarL2:= 0 Else VarL2:=
VarL2-MaxVar;
end;

Procedure CheckState;
begin; Assign(st,'State.LMT'); Reset(st);
      Readln(st,MAX);
      Readln(st,STD);
      For i:= 1 to 3 do Readln(st,P[i]);
      For i:= 1 to 4 do Readln(st,State[i]);
      Readln(st,Slope);
      Readln(st,LastDay);
      Readln(st,E5);
      Readln(st,MaxVar);
      Close(st);

      CurSlope:= ((Result[2]*5+Result[1]) -
(Result[2]*1+Result[1]));
      CurLastDay:= Result[2]*5+Result[1];
      top:= 0;
      For i:= 1 to 5 do top:= top+(Result[2]*i+Result[1]);
      CurE5:= top;
      R_Slope:= State[4];
      For i:= 1 to 3 do
        If CurSlope >= (P[i]-6) Then begin; R_Slope:= State[i];
                                          i:= 3; end;

      R_LastDay:= State[4];
      For i:= 1 to 3 do
        If CurLastDay >= P[i] Then begin; R_LastDay:= State[i];

```



```

                                                    i:= 3; end;
R_E5:= State[4];
For i:= 1 to 3 do begin; top:= 0;
  For j:= 1 to 5 do top:= top+((P[i]-6)*(j-1)/4)+6;
  If CurE5 >= top Then begin; R_E5:= State[i];
                                                    i:= 3; end; end;
  For i:= 1 to 3 do
    Limit[i]:=
      (State[i+1]*(Slope+LastDay)+State[i]*E5)/State[1];
    FinalValue:=
      (R_Slope*Slope+R_LastDay*LastDay+R_E5*E5)/State[1];
    If FinalValue > Limit[1] Then
RealState:= 'Severe Problem';
    If (FinalValue <= Limit[1]) and (FinalValue > Limit[2]) Then
RealState:= 'Problem';
    If (FinalValue <= Limit[2]) and (FinalValue > Limit[3]) Then
RealState:= 'Potential Problem';
    If (FinalValue <= Limit[3]) Then
RealState:= 'Optimum Operations';
    Variance;
    Violation:= 0;
    For i:= 1 to n do
      If (EsasY[i] > (Result[2]*EsasX[i]+Result[1]+MaxVar)) or
        (EsasY[i] < (Result[2]*EsasX[i]+Result[1]-MaxVar))
Then
      Violation:= Violation+1;
    If ((RealState= 'Optimum Operation') or
      (RealState= 'Potential Problem')) and
      (Violation >= 4) Then RealState:= 'Problem';
  end;
Procedure Statistics;
begin; Ey2:= 0; Ey:= 0;
  For i:= 1 to n do begin; Ey2:= Ey2+Y[i]*Y[i];
    Ey := Ey +Y[i]; end;
  Ey:= Ey/n; Std1:= Sqrt(Ey2/n-Ey*Ey);
  For i:= 1 to n do begin;
    If Ftype in [2,4] Then Fx[i]:= ln(X[i])
      Else Fx[i]:= X[i];
    Func:= 0;
    For j:= 1 to (Order+1) do Func:=
Func+Result[j]*Power(Fx[i],j-1);
    If Ftype in [3,4] Then Fy[i]:= Exp(Func)
      Else Fy[i]:= Func;
  end;
  Ey2:= 0; Ey:= 0;
  For i:= 1 to n do begin; Ey2:= Ey2+Fy[i]*Fy[i];
    Ey := Ey +Fy[i]; end;
  Ey:= Ey/n; Std2:= Sqrt(Ey2/n-Ey*Ey);

```

```

        If Std2 > Std1 Then R:= Std1/Std2
            Else R:= Std2/Std1; D:= R*R*100;
    end;

Procedure GenerateFunction;
    begin; Increment:= (Xmax-Xmin)/50;
        Fx[1]:= Xmin;
        Func:= 0;
        For j:= 1 to (Order+1) do
            If Ftype in [2,4] Then Func:=
Func+Result[j]*Power(ln(Fx[1]), j-1)
            Else Func:= Func+Result[j]*Power(
Fx[1] ,j-1);
            If Ftype in [3,4] Then Fy[i]:= Exp(Func)
                Else Fy[i]:= Func;

        For i:= 2 to 51 do begin;
            Fx[i]:= Fx[i-1]+Increment;
            Func:= 0;
            For j:= 1 to (Order+1) do
                If Ftype in [2,4] Then Func:=
Func+Result[j]*Power(ln(Fx[i]), j-1)
                Else Func:= Func+Result[j]*Power(
Fx[i] ,j-1);
                If Ftype in [3,4] Then Fy[i]:= Exp(Func)
                    Else Fy[i]:= Func;

            If Fy[i] > Ymax Then Ymax:= Fy[i];
            If Fy[i] < Ymin Then Ymin:= Fy[i];
        end;
    end;

Procedure DrawLine(x1,y1,x2,y2: Integer; Color:Byte);
    begin; XX[1]:= x1; YY[1]:= y1;
        XX[2]:= x2; YY[2]:= y2;
        DrawXY(XX,YY,2,Color);
    end;

```

```

BEGIN
    Input;
    Curve(X,Y,n,Ftype,Order);
    CheckState;
    Statistics;
    GenerateFunction;

    Assign(dt,'STATE.VAL'); Rewrite(dt);
    Writeln(dt,''+RealState+'');
    Close(dt);

    Init;
    SetColor(Red);
    SetBkColor(Blue);
    SetTextStyle(1,Horizdir,3);

```

```

    Msg:= 'Effluent Nitrification Data';
    OutTextxy(Trunc(GetMaxX/2-TextWidth(Msg)/2),TextHeight(Msg),Msg);
    SetColor(LightCyan);
    SetTextStyle(1,Horizdir,2);
    Msg:= 'Nitrification State: '+RealState;

    OutTextxy(Trunc(GetMaxX/2-TextWidth(Msg)/2),(3*TextHeight(Msg)),Msg);
    SetColor(Yellow);
    SetTextStyle(0,0,0);
    Str(R:5:3,Msg);
    Msg:= 'Correlation Coeffecient = '+Msg;

    OutTextxy(Trunc(GetMaxX/2-TextWidth(Msg)/2),(10*TextHeight(Msg)),Msg);
    Str(((n-Violation)/n*100):3:0,Msg);
    Msg:= Msg+'% of data within the Variance Range';

    OutTextxy(Trunc(GetMaxX/2-TextWidth(Msg)/2),(11*TextHeight(Msg)),Msg);

    SetColor(GetMaxColor);
    Xaxis(Xmin,Xmax,AraX,'Today (Day 5) & Last 4 days',Font,Size);
    Yaxis(0,Max-10,10,'Effluent NH4+ Concentration
(mg/L)',Font,Size);

    DrawXYPoint(EsasX,EsasY,n,5,2,LightBlue);
    DrawLine(1,6,5,P[1],DarkGray);
    DrawLine(1,6,5,P[2],DarkGray);
    DrawLine(1,6,5,P[3],DarkGray);

    SetLineStyle(DashedLn,0,ThickWidth);
    DrawLine(0,STD,6,STD,LightRed);

    SetLineStyle(SolidLn,0,NormWidth);
    DrawXy(Fx,Fy,51,LightCyan);
    SetLineStyle(DottedLn,0,NormWidth);
    DrawLine(1,VarH1,5,VarH2,LightCyan);
    DrawLine(1,VarL1,5,VarL2,LightCyan);

    Getkey;
    ClearDevice;
    CloseGraph;
    RestoreCrtMode;
END.

*****
Unit Axis;

Interface
Uses Menu4,Crt,Graph;

Type AnyString = String[80];
Var
    XaxisMin, XaxisMax, YaxisMin, YaxisMax: Real;
    XaxisLong, YaxisLong, Value, Littletic: Real;
    PointX, PointY : Integer;

```

```

    Point1X, Point1Y, Point2X, Point2Y, Point3X, Point3Y: Integer;
    ij, factor, tic, locX, locY, locValue, LocLittleX, LocLittleY:
Integer;
    s: String[6];

PROCEDURE Xaxis(min,max: Real; ara: Integer;
    Xaxislabel: Anystring;
    Font, Size: Word);
PROCEDURE Yaxis(min,max: Real; ara: Integer;
    Yaxislabel: Anystring;
    Font, Size: Word);
PROCEDURE DrawXY (X, Y: Realarray; m: Integer; Renk: Byte);
PROCEDURE DrawXYBar (X, Y: Realarray; m: Integer; Renk: Byte);
PROCEDURE DrawXYPoint(X, Y: Realarray; m, Shape, Size: Integer; Renk:
Byte);

```

Implementation

```

PROCEDURE Xaxis(min,max: Real; ara: Integer;
    Xaxislabel: Anystring;
    Font, Size: Word);
Label bastan;
Procedure Pointdata;
    begin; Point1X:= Trunc( 2.5*GetmaxX/20);
    Point1Y:= Trunc( 1.0*GetmaxY/10);
    Point2X:= Trunc( 2.5*GetmaxX/20);
    Point2Y:= Trunc( 8.0*GetmaxY/10);
    Point3X:= Trunc(18.5*GetmaxX/20);
    Point3Y:= Trunc( 8.0*GetmaxY/10);
    XaxisLong:= 16*GetmaxX/20;
    YaxisLong:= 7*GetmaxY/10;
    end;
Procedure Xlabel;
    begin; SetTextJustify(CenterText,CenterText);
    PointX:= Trunc(Point2X+(Point3X-Point2X)/2);
    PointY:= Trunc(Point2Y+GetmaxY/10);
    OutTextxy(PointX,PointY,Xaxislabel);
    end;

    begin; Pointdata;
    XaxisMin:= min-((max-min)/0.9)*0.05;
    XaxisMax:= max+((max-min)/0.9)*0.05;
    factor:= 0;
    tic:= Trunc((XaxisMax-XaxisMin)/ara);
bastan: Case tic Of
    0..1: tic:= 1;
    1..3: tic:= 2;
    3..8: tic:= 5;
    else begin; factor:= Succ(factor);
        tic:= Trunc(tic/10); Goto bastan; end;
    end;
tic:= Trunc(tic*exp(factor*Ln(10)));
XaxisMin:= (Trunc(XaxisMin/tic))*tic;
XaxisMax:= (Trunc(XaxisMax/tic))*tic+tic;

```

```

    If XaxisMin < 0 Then XaxisMin:= 0;
    ij:= 0; Value:= XaxisMin-tic;
    Line(Point2X,Point2Y,Point3X,Point3Y);
    SetTextJustify(CenterText,CenterText);
    SetTextStyle(Font,Horizdir,Size);
    Repeat
        ij:= Succ(ij);
        Value:= Value+tic;
        locX:=
Trunc(Point2X+(XaxisLong*tic/(XaxisMax-XaxisMin))*(ij-1));
        Str(Value:0:0,s);
        locValue:= locX;
        If locX <= Point3X Then begin;
        Line(locX,Point2Y,locX,Point2Y+Trunc(GetmaxY/100));
        OutTextxy(locValue,Point2Y+Trunc(GetmaxY/30),s); end;

    Until (locX > Point3X) or (Value > XaxisMax+tic);
    Case tic of
        0 .. 11: If ara < 10 Then Littletic:= tic/10*ara
                Else Littletic:= tic/10;
        11 .. 51: If ara < 5 Then Littletic:= tic/5*ara
                Else Littletic:= tic/5;
                Else Littletic:= tic/2;
    end;
    ij:= 0;
    Repeat
        ij:= Succ(ij);
        LocLittleX:=
Trunc(Point2X+(XaxisLong*Littletic/(XaxisMax-XaxisMin))*(ij-1));

    Line(LocLittleX,Point2Y,LocLittleX,Point2Y+Trunc(GetmaxY/125));
    Until LocLittleX > Point3X;
    SetColor(Blue);

    Line(LocLittleX,Point2Y,LocLittleX,Point2Y+Trunc(GetmaxY/125));
    SetColor(GetMaxColor);
    Xlabel;

    end;

PROCEDURE Yaxis(min,max: Real; ara: Integer;
                Yaxislabel: Anystring;
                Font, Size: Word);
Label bastan;
Procedure Ylabel;
    begin; SetTextJustify(CenterText,CenterText);
           SetTextStyle(Font,Vertdir,Size);
           PointX:= Trunc(Point2X-GetmaxX/10);
           PointY:= Trunc(Point1Y+(Point2Y-Point1Y)/2);
           OutTextxy(PointX,PointY,Yaxislabel);
           SetTextStyle(Font,Horizdir,Size);

    end;

begin; YaxisMin:= min-((max-min)/0.9)*0.05;
       YaxisMax:= max+((max-min)/0.9)*0.05;

```

```

        factor:= 0;
        tic:= Trunc((YaxisMax-YaxisMin)/ara);
    bastan:  Case tic Of
            0..1: tic:= 1;
            1..3: tic:= 2;
            3..8: tic:= 5;
            else begin; factor:= Succ(factor);
                tic:= Trunc(tic/10); Goto bastan; end;
        end;
        tic:= Trunc(tic*exp(factor*Ln(10)));
        YaxisMin:= (Trunc(YaxisMin/tic))*tic;
        YaxisMax:= (Trunc(YaxisMax/tic))*tic+tic;
        If YaxisMin < 0 Then YaxisMin:= 0;
        ij:= 0; Value:= YaxisMin-tic;
        Line(Point1X,Point1Y,Point2X,Point2Y);
        SetTextJustify(RightText,CenterText);
        SetTextStyle(Font,Horizdir,Size);
        Repeat
            ij:= Succ(ij);
            Value:= Value+tic;
            locY:=
Trunc(Point2Y-(YaxisLong*tic/(YaxisMax-YaxisMin))*(ij-1));
            Str(Value:0:0,s);
            locValue:= Trunc(Point2X-GetmaxX/50);
            If locValue < 0 Then locValue:= 0;
            If locY >= Point1Y Then begin;
                Line(Point2X,locY,Point2X-Trunc(GetmaxX/100),locY);
                OutTextxy(locValue,locY,s); end;

        Until (locY < Point1Y) or (Value > YaxisMax+tic);
        Case tic of
        0 .. 11: If ara < 10 Then Littletic:= tic/10*ara
                Else Littletic:= tic/10;
        11 .. 51: If ara < 5 Then Littletic:= tic/5*ara
                Else Littletic:= tic/5;
        Else Littletic:= tic/2;
        end;
        ij:= 0;
        Repeat
            ij:= Succ(ij);
            LocLittleY:=
Trunc(Point2Y-(YaxisLong*Littletic/(YaxisMax-YaxisMin))*(ij-1));
        Line(Point2X,LocLittleY,Point2X-Trunc(GetmaxX/220),LocLittleY);
        Until LocLittleY <= Point1Y;
        Ylabel;
    end;

PROCEDURE DrawXY(X, Y: Realarray; m: Integer; Renk: Byte);
    begin; SetColor(Renk);

Moveto(Trunc(Point2X+(XaxisLong*(X[1]-XaxisMin)/(XaxisMax-XaxisMin))),
Trunc(Point2Y-(YaxisLong*(Y[1]-YaxisMin)/(YaxisMax-YaxisMin))));

```

```

        ij:= 0;
        Repeat
            ij:= Succ(ij);
            locX:=
Trunc(Point2X+(XaxisLong*(X[ij]-XaxisMin)/(XaxisMax-XaxisMin)));
            locY:=
Trunc(Point2Y-(YaxisLong*(Y[ij]-YaxisMin)/(YaxisMax-YaxisMin)));
            Lineto(locX,locY);

        Until ij > m-1;
    end;

PROCEDURE DrawXYBar(X, Y: Realarray; m: Integer; Renk: Byte);
    begin;
Moveto(Trunc(Point2X+(XaxisLong*(X[1]-XaxisMin)/(XaxisMax-XaxisMin))),
Trunc(Point2Y-(YaxisLong*(Y[1]-YaxisMin)/(YaxisMax-YaxisMin))));
        ij:= 0;
        Repeat
            ij:= Succ(ij);
            locX:=
Trunc(Point2X+(XaxisLong*(X[ij]-XaxisMin)/(XaxisMax-XaxisMin)));
            locY:=
Trunc(Point2Y-(YaxisLong*(Y[ij]-YaxisMin)/(YaxisMax-YaxisMin)));
            locX:= locX-1;      { ortala }
            SetColor(Yellow);
            Rectangle(locX-2,Point2Y,locX+2,locY-1);
            SetFillStyle(SolidFill,Renk);
            Bar(locX-1,Point2Y,locX+1,locY);

        Until ij > m-1;
    end;

PROCEDURE DrawXYPoint(X, Y: Realarray; m, Shape, Size: Integer; Renk:
Byte);
    begin; SetColor(Renk);
        ij:= 0;
        Repeat
            ij:= Succ(ij);
            locX:=
Trunc(Point2X+(XaxisLong*(X[ij]-XaxisMin)/(XaxisMax-XaxisMin)));
            locY:=
Trunc(Point2Y-(YaxisLong*(Y[ij]-YaxisMin)/(YaxisMax-YaxisMin)));
            Case Shape of
                1: Rectangle(locX-Size,locY-Size,locX+Size,locY+Size);
                2: begin; Line(locX-Size,locY-Size,locX+Size,locY+Size);
                    Line(locX-Size,locY+Size,locX+Size,locY-Size);
                end;
                3: Circle(locX,locY,Size);
                4: begin; SetFillStyle(SolidFill,Renk);
                    Bar(locX-Size,locY-Size,locX+Size,locY+Size);
                end;
                5: begin; Line(locX-Size,locY-Size,locX+Size,locY+Size);
                    Line(locX-Size,locY+Size,locX+Size,locY-Size);
            end;
        end;
    end;

```

```

Line(locX-Size,locY      ,locX+Size,locY      );
Line(locX      ,locY+Size,locX      ,locY-Size);
end;
      end;
    Until ij > m-1;
  end;
END.

```

```

*****
{$R-}   {Range checking off}
{$B+}   {Boolean complete evaluation on}
{$S+}   {Stack checking on}
{$I+}   {I/O checking on}
{ $M 65500,0,655360} {Turbo 3 default stack and heap}

```

Unit Curvefit;

Interface

Uses Menu4;

Var Result: Array [1..8] of Real;

```

Function Power(a, b: Real): Real;
PROCEDURE Curve(X, Y: RealArray; n, Eqtype, Order: Integer);

```

Implementation

```

Function Power(a, b: Real): Real;
  begin; If a= 0 Then If b= 0 Then Power:= 1
                    Else Power:= 0
                    Else Power:= (a/ABS(a)) * exp(b * ln(ABS(a)));
  end;

```

```

PROCEDURE Curve(X, Y: RealArray; n, Eqtype, Order: Integer);
Var i, j, k, m, q, z: Integer;
    t: Real;
    Ex, IEx: Array [1..8,1..16] of Real;
    Exy: Array [1..8] of Real;

```

```

Procedure Invert;
  begin; i:= 0;
        Repeat
          i:= Succ(i);
          j:= (Order+1)+1-1;
          Repeat
            j:= Succ(j);
            If i= j-(Order+1) Then Ex[i,j]:= 1
                               Else Ex[i,j]:= 0;
          Until j >= (2*(Order+1));
        Until i >= (Order+1);

        k:= 0;

```



```

Repeat
  k:= Succ(k);
  j:= k-1;
  Repeat
    j:= Succ(j);
    IEx[k, j]:= Ex[k, j]/Ex[k, k];
  Until j >= (2*(Order+1));
  i:= 0;
  Repeat
    i:= Succ(i);
    If i= k then i:= k+1;
    m:= 0;
    Repeat
      m:= Succ(m);
      IEx[i, m]:= Ex[i, m]+Ex[i, k]*-IEx[k, m];
    Until m >= (2*(Order+1));
  Until i >= (Order+1);
  q:= 0;
  Repeat
    q:= Succ(q);
    z:= 0;
    Repeat
      z:= Succ(z);
      Ex[q, z]:= IEx[q, z];
    Until z >= (2*(Order+1));
  Until q >= (Order+1);
Until k >= (Order+1);

end;

( Begin Curvefit )
begin; For i:= 1 to (Order+1) do For j:= 1 to (2*(Order+1)) do
Ex[i, j]:= 0;
For i:= 1 to (Order+1) do begin; Result[i]:= 0; Exy[i]:= 0;
end;
Case Eqtype of
  1: ;
  2: For i:=1 to n do If X[i] > 0 Then X[i]:= ln(X[i])
Else begin; ErrorMsg('Data
contains 0 or negative terms ');
Halt; end;
  3: For i:=1 to n do If Y[i] > 0 Then Y[i]:= ln(Y[i])
Else begin; ErrorMsg('Data
contains 0 or negative terms ');
Halt; end;
  4: For i:=1 to n do If (X[i] > 0) or (Y[i] > 0)
Then begin; X[i]:= ln(X[i]); Y[i]:=
ln(Y[i]); end
Else begin; ErrorMsg('Data contains 0 or
negative terms ');
Halt; end;
end;
For i:= 1 to (Order+1) do
For j:= 1 to (Order+1) do

```

```

    For k:= 1 to n do begin;
        t:= X[k]; Ex[i,j]:= Ex[i,j]+Power(t,(i+j-2)); end;
Ex[ 1, 1]:= n;
Invert;
For i:= 1 to (Order+1) do
    For k:= 1 to n do begin;
        t:= X[k]; Exy[i]:= Exy[i]+Power(t,(i-1))*Y[k]; end;
For i:= 1 to (Order+1) do
    For j:= ((Order+1)+1) to (2*(Order+1)) do
        Result[i]:= Result[i] + Exy[j-(Order+1)] * IEx[i,j];
end;
END.

```