UNIVERSITÉ DU QUÉBEC

THÈSE PRÉSENTÉE À
L'UNIVERSITÉ DU QUÉBEC À CHICOUTIMI
COMME EXIGENCE PARTIELLE
DU DOCTORAT EN INGÉNIERIE

par

SONG PING LU

CONTRÔLE ET SUPERVISION DU PROCÉDÉ D'ÉLECTROLYSE DE L'ALUMINIUM PAR SYSTÈME EXPERT

Control and Supervision of the Aluminium Electrolysis Process
with Expert System

SEPTEMBRE 2002

RÉSUMÉ

La cuve d'électrolyse est l'élément central dans la réduction de l'aluminium. En dépit des systèmes de contrôle automatique appliqués sur l'opération des cuves, une quantité significative d'informations sur leurs états n'est pas encore utilisée dans le processus décisionnel. D'ailleurs, la qualité de décision dépend bien souvent de l'opérateur responsable.

Ce système expert à base de règles à deux niveaux est construit à partir de l'expertise disponible des opérateurs et de celle des ingénieurs du procédé d'électrolyse. Ce système est conçu pour diagnostiquer autant les cuves de type général que celles de types particuliers. De plus, il peut fonctionner en mode autonome tout en utilisant des données d'entrées locales à la station de travail ou en mode réseau en utilisant les données du procédé réel comme valeur d'entrée. Dans l'architecture réseau, le procédé réel peut être remplacé par un simulateur de cuve (un modèle mathématique) utilisant des mécanismes de transfert d'information semblable au système d'acquisition de données des procédés en temps réel. Cela permet de tester explicitement les tâches du système expert sur la surveillance du procédé et ses alarmes. L'agencement a comme objectif de proposer une aide aux opérateurs pour créer des analyses détaillées de l'état des cuves, de détecter la présence de défaut dans le procédé, de faire l'analyse de tendance et de proposer l'affectation de cible à long terme. Il peut également être

présenté à l'ingénieur de contrôle comme référence pour le réglage de points de consigne sur certains régulateurs pertinents.

L'architecture de la base de connaissance est conçue de manière à permettre la distribution de l'application aux divers types de cuves afin de simplifier la mise à jour éventuelle du système. C'est pour cette raison que la structure de la base de connaissances et la stratégie de raisonnement sont conçues avec des caractères uniques.

Cette thèse fournit l'ensemble de la connaissance saisie au sujet du procédé d'électrolyse de l'aluminium et des secteurs appropriés. Celle-ci comprend la connaissance générale du domaine pour l'ingénierie cognitive aussi bien que la connaissance spéciale pour les types particuliers de cuves. Elle décrit également la construction du système expert et montre quelques exemples accompagnés de discussions détaillées sur différents cas de diagnostique.

ABSTRACT

The electrolytic cell is the central element of the aluminum electrolysis. In spite of the automatic control systems applied to run the aluminum electrolytic cells, a significant amount of information about the status of the cells is still not involved in the decision making process. Moreover, the quality of the decision depends on the operator in charge.

A two-level rule-based expert system is built which incorporates the available "operator" and "engineer" expertise. This expert system is designed to serve both a generalized as well as a particular type of cell. It can work in either off-line or on-line mode. When in on-line mode, the process data can be accessed via a network and a cell simulator can also be connected to test the expert system-based process monitoring and alarm management. The proposed expert system assists the operators to make a detailed analysis of the cell's state, to detect faults, and to conduct trend analysis or a long-term target assignment. The expert system can also be a reference for the adjustment of the set points by control system engineers.

The knowledge base design is intended to expand the application to the various types of cells and to facilitate a system update. For this reason, the

structure of the knowledge base and the reasoning strategy are designed with some unique characters.

This thesis provides the acquired knowledge about the aluminum electrolysis process and relevant areas. It consists of the general domain knowledge for the knowledge engineering as well as the special knowledge for particular types of cells. It also describes the construction of the expert system and shows some examples with detailed discussions of the cell diagnosis.

ACKNOWLEDGMENTS

I am greatly indebted to my research director, Dr. Rung Tien Bui, professor of Université du Québec à Chicoutimi and director of the CRSNG-ALCAN Chair in process engineering and co-director Dr. Laszlo Tikasz, professor of Université du Québec à Chicoutimi, for their guidance, advice, support and encouragement. Their extensive knowledge helps me shape my way of thinking.

I would like to express my gratitude to Mr. Alton T. Tabereaux of Alcoa Inc., for his permission to use the knowledge in his course materials. Special thanks go also to Dr. Janos Horvath, professor of Technical University of Miskolc and visiting professor of Université du Québec à Chicoutimi, for his very important contribution from his extensive knowledge and very helpful conversations.

I am indebted to Dr. Vinko Potocnik of Arvida Research and Development Center of ALCAN for his contributions and helpful advice.

I would like to acknowledge all the members of LECAP, for their collaboration.

Finally, I would like to express my sincere thanks for the encouragement and support from my parents and my family.

CONTENTS

LIST OF TABLES

LIST OF FIGURES

CHAPTER 1 INTRODUCTION

1.1 Problems addressed

The electrolytic cell is the central component of the aluminum electrolysis. In spite of the automatic control systems applied worldwide to run the aluminum electrolytic cells, a significant part of information about the status of the cell is still not involved in the decision making process (e.g. visual observations and additional measurements made by the operator). Moreover, the quality of the decision depends on the operator in charge.

In smelters, the cell operating status depends upon many factors, such as cell design, cell operation history, raw material properties, process control system, routine maintenance, operation practice, performance of tools and utilities. But even in modern plants, the basic characteristics of the process control system are still the same: only the line current and the cell voltage are measured continuously while other process data are measured and logged only occasionally [1]. Yet, many automated control actions, such as alumina feeding, anode effect treatment, metal tapping, fluoride addition, anode changing and sludge removal strongly depend on continuous measurements. In this thesis, the use of an expert system is therefore proposed to support the cell operator.

Expert systems are applications of artificial intelligence to a certain field of practice. The program employed in the expert system is a computer algorithm, which performs human-like reasoning, often using rules for the interpretation of the given knowledge. In practice, an expert system is developed using software tools called shells. These shells commonly contain inference mechanisms (backward chaining, forward chaining, or both), program editors, and a user interface. But they do not contain any specific domain-knowledge and process data. The relevant knowledge is required to be entered in a specified format.

Expert system technology has been developing rapidly over the last two decades and has been applied in many fields. However, the number of real, operational, industrial applications is still quite limited. One of the major reasons is that development of expert systems is largely based today on empirical methods and is not supported by general methodologies. It is more like handicraft than engineering [2]. Such conditions are often considered as a “bottleneck problem” of expert system applications, even found in aluminum industrial applications.

It is obvious that all kinds of aluminum electrolysis cells have common properties and operate on the same fundamental theory, and have similar structures and process control systems. The general process knowledge is abstracted from such common properties.

Most of the existing cell expert systems are designed for particular purposes. The different aluminum plants have their particular type of cell with a special structure, geometry and operational technique meaning each particular type of cell has its own properties and characteristics. Thus, it is impossible to use only one knowledge base to suit all the particular cells, due to their different properties. Therefore, the applicable knowledge, coded in expert systems, has to present the specialties.

In the real industrial process, using only one, generalized expert system for various kinds of cells would not be economical. Much work is needed to capture the specialties of each particular cell type. This is one of the cardinal problems that affect the structure of expert systems applied to the aluminum industry. Finding a general methodology, which will comprise the knowledge of two most common cell types effectively, is the main task of our design.

1.2 Objectives

The objectives of this project are the following: First, construct a knowledge base, which incorporates the expertise of cell operation. Second, cast this knowledge into an expert system. Third, process this knowledge and make decisions about the cell. Fourth, connect the expert system to a simulator through networking to realize on-line diagnosis. The simulator acts as a virtual cell and allows us to diagnose the problems of the electrolysis process and to compare traditional control with expert system based process monitoring and alarm management. The proposed arrangement will aid the operators in:

- Detailed analysis of the cell's state
- Fault detection
- Process control assistance
- Trend analysis.

With such objectives in mind, we designed an expert system named ***Aluminum Electrolysis Process Expert System*** (AEPES) consisting of two sub-systems: ***Engineer Expert System*** (ENGES) and ***Operator Expert System*** (OPEES). The ENGES incorporates the more general knowledge of the aluminum electrolysis process but does not focus on any particular cell, in contrast to the OPEES, which adds the particular knowledge on a specific type of cell. As the general knowledge is applicable to all types of cells and is essentially knowledge in process metallurgy, it is mainly applied to the construction of the ENGES.

The general and particular knowledge are modularized, and embedded into the knowledge bases. This methodology allows us to easily develop expert systems for different types of cells. The relationship between these two expert systems can be compared to that between the tasks of an engineer and an operator. The ENGES is concerned with general, theoretical analysis, whereas the OPEES acts like a skilled operator and performs consultation and makes process-related decisions.

The OPEES can be used in on-line mode to realize real-time consultations. The applied expert system shells have a communication bridge, which allows access to process data through the network. Then the OPEES will examine the state of the cell based on the knowledge and "fresh" plant data. At the first, development stage, a simulator is used instead of the real process to reduce the need for the time-consuming plant test. Based on the first stage testing results, the expert system is improved to become more stable and reliable. Then direct access to the real process (through the network) to realize real-time consultation and trend control supervision is the goal of the second stage of the work.

The AEPES is built using two shells, Comdale/X and Gensym G2. The former is easy to handle and economical for smaller expert system development projects. The latter is more powerful in managing and optimizing large-scale applications and is suited for complex systems. The two shells incorporate the same knowledge and the structure of the knowledge base is identical. The user interface and data communication bridges differ. The Gensym G2 provides a powerful development environment, which allows us to improve the efficiency of the real-time performance of AEPES.

1.1 Organization of the thesis

In Chapter 2, the process fundamentals of the aluminum electrolysis and the relevant domain knowledge are introduced. We analyze the main aspects of this complex process and focus on the important parts required for the knowledge base building. The production technology of the different types of cell is also introduced.

In Chapter 3, the fundamental concepts of artificial intelligence technology and expert system are introduced. The emphasis is on the rule-based expert system, which is used in our project. In the last section, we analyze the reasons for using the expert system for aluminum electrolysis process.

In Chapter 4, a review of industrial applications of the expert system is given. We examine the available literature on industrial applications of the expert system, especially for aluminum electrolysis process. The different characters of each system are analyzed that could benefit our system design. The lack of literature indicates that additional research remains to be done in this area.

In Chapter 5, we discuss the knowledge acquisition in two parts. Part I is for the knowledge engineering, and Part II is directly applied for the knowledge base casting. Both are necessary to integrate knowledge for our expert system building. In Part II, some domain knowledge collected from open sources is introduced in detail. They are fundamental for our knowledge base.

Chapter 6 recounts the basic design of the AEPES, which is developed using Comdale/X. Two sub-systems are introduced; they are designed for different purposes. We give a detailed description of the structure of the knowledge base, user interface design, operation characteristics and knowledge base maintenance. Several technologies are discussed for the improvement of the reasoning process. Also, some examples can be found at the end of this chapter.

In Chapter 7, we introduce the advanced design of AEPES, which is coded in Gensym G2. This powerful expert system shell allows us to create enhanced on-line consultation.

In Chapter 8, the general conclusions of the project and a list of recommendations for future work are given.

In Appendix 1, applied expert system shells are briefly introduced with two application examples. In Appendix 2, the VS-ANODE expert system is analyzed. In Appendix 3, several improvements of the reasoning process are described and concluded. In Appendix 4, the evaluation testing results are discussed. The flowchart of AEPES operation can be found in Appendix 5. The tutorials of ENGES and OPEES are given in Appendix 6 and Appendix 7.

CHAPTER 2 THE ALUMINUM ELECTROLYSIS PROCESS

2.1 Aluminum

Aluminum is the most abundant metal in the earth's crust (7.5 wt.%). It is now the second most widely used metal after steel. An excellent conductor of heat and electricity, aluminum has other superior properties, such as being lightweight, as well as being versatile and strong when alloyed with other metals, and resistant to atmospheric corrosion. The plentiful reserves and its superior properties allow aluminum to be widely applied in aviation, mechanical, chemical, electrical, architectural and metallurgical industries **[1]**. By using traditional methods, aluminum in metal form is very difficult to obtain. In 1886, Charles Martin Hall and Paul Louis Toussaint Héroult simultaneously and independently developed the process to make aluminum metal, which is called the Hall-Héroult process. The original concept of this process remains in use today.

2.2 Brief description of the aluminum electrolysis process

2.2.1 Principle of the Hall-Héroult process

The fundamental concept of aluminum production is using electricity to reduce alumina (Al_2O_3) to aluminum (Al). The melting point of alumina is close to 2030 °C, which is in general too high for commercial production. In order to produce aluminum more efficiently, i.e. at a lower operating temperature, Charles Hall and Paul Héroult found cryolite (Na_3AlF_6) as chemical compound that acted as a solvent for alumina. In the Hall-Héroult process, the alumina is dissolved in the molten cryolite, which reduces the temperature to 950-980 °C.

Actually, in the electrolytic cell, the alumina is dissolved in a bath, which is typically composed of cryolite (Na_3AlF_6), with small percentage of AlF_3 and CaF_2. The anode is made of carbon (C). During the electrolysis process, the aluminum ions are reduced to aluminum metal and collect at the bottom of the cell. The oxygen is discharged at the carbon anode, where it reacts with carbon to form carbon dioxide. The primary chemical reaction is given by the equation:

$$2Al_2O_3 + 3C = 4Al = 3CO_2 \qquad (2\text{-}1)$$

This equation determines the electrochemical decomposition potential of alumina electrolyzed with carbon anodes. However, in this process, some of the metal, which dissolved in the electrolyte, can be reoxidized by CO_2 evolved at the anode according to secondary reaction:

$$Al + CO_2 = Al_2O_3 + CO \qquad (2\text{-}2)$$

The secondary reaction leads to decrease the metal current efficiency and increase carbon monoxide in the anode gas and overall carbon consumption. [2, 3]

2.2.2 Industrial aluminum electrolysis

Although the original concept of the Hall-Héroult process has not changed, improvements have been made in a continuing effort to lower production costs in both equipment and materials for today's industrial production. The electrolytic, or smelting, process takes place in electrolytic cells, of which there may be several hundreds or more in a modern plant.

There are two main types of aluminum smelting technology: the Söderberg technology and the prebake anode technology. The principal difference between them is the type of anode used. The Söderberg technology uses one continuous anode, which bakes in situ from anode paste. The prebake technology uses multiple anodes in each cell, which are prebaked. The basic structure of modern electrolytic cells, whether a Söderberg or a prebake anode, consists of a rectangular steel shell; that is lined with thermally insulating refractory materials. Inside the cell, there is an inner lining of prebaked carbon cathode blocks with embedded steel current collector bars where the molten aluminum is in direct contact with the carbon blocks. Carbon sidewalls and thermal insulating materials, such as alumina powder or refractory bricks, complete the cathode construction. Today's prebake cells commonly include a hooding that is connected to a gas exhaust and scrubbing system. The general structure of the aluminum electrolytic cell is given in Figure 2.1.

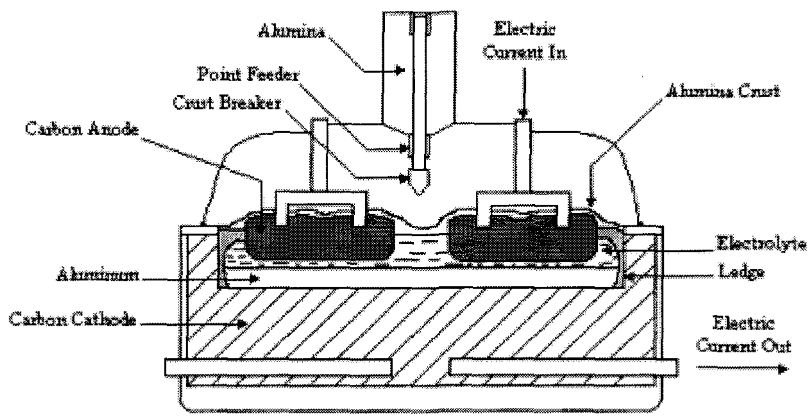

Figure 2.1 General structure of a prebake anode cell

Generally, a Söderberg anode or two rows of prebake anodes are suspended from bus bars, which carry the electric current. Alumina is added from supply bins above the cells, or from a truck. The alumina is fed to the bath periodically by breaking the crust in the center between two rows of prebake anodes or at the sides of a cell. Finally, aluminum is removed (tapped) from the cells at regular intervals and sent to a holding furnace of the cast house from which various forms of ingots are cast.

The cells are placed and connected in series, which are called potlines. The process requires direct current, which ranges from 50 to 500 kA, depending on the technology used. The cell voltage is normally from 4 to 5 V. The direct electric current is passed through the bath from the suspended anodes to the cathodes, to the bottom of the carbon lining of the cells, and on to the collector bars that are embedded in the bottom of the carbon lining. The electric energy required to produce aluminum is high: 13 – 17 kWh/kg Al.

2.3 Relevant elements of the Hall-Héroult process

The knowledge acquisition of aluminum electrolysis process is related to many relevant elements of the routine operations of the Hall-Héroult process. A better understanding of these elements is helpful in building our knowledge base. In making use of this domain knowledge, more accurate diagnosis and better suggestions will be possible. A brief description of the relevant concepts is given below [4, 5, 6].

Electrolyte (bath)

The composition of the electrolyte may be expressed as the bath ratio (mass ratio of NaF and AlF_3). The bath ratio is an important factor of the cell operation.

Alumina

In the aluminum electrolysis process, alumina is the basic raw material for metal production. It is also used as thermal insulation on top of the anodes or the crust. In prebake technology, the anode cover also protects anode from air oxidation. Alumina on the crust also absorbs HF, an important pollutant. Therefore the quality of alumina added to the electrolyte and the quantity of alumina cover are of great importance for cell operation. The specifications imposed by a smelter include such factors as purity, bulk and pycnometric densities and particle size distribution. Purity is one of the specifications of great importance to the quality of the aluminum and the performance of the cell. The main impurities in liquid aluminum are iron and silicon. They are monitored by sampling at regular intervals.

Alumina concentration

The alumina concentration in the cell depends on the amount and the frequency of the break-and-feed that is required to maintain it within a limited range. If too much alumina is fed, “sludge” is deposited on the top of cathode carbon block, whereas insufficient feeding results in an anode effect. The cell voltage is used to monitor and control the alumina concentration so that there is little sludge and few anode effects. In practice, the cell voltage is substituted by a converted parameter, so-called pseudo-resistance (See **Section 5.2.3.1 “Typical process control technology”**).

Anode effect

The anode effect occurs when the alumina concentration is below 1 – 1.5%. Typically, during the anode effect cell voltage suddenly increases to 20 – 50 volts. The negative features of the anode effect are: an increase in energy consumption, a reduction in metal production, a higher rate of emissions and the overheating of the cell causing an unstable operation condition. The anode effect has two positive features: it provides a control of the alumina content in the electrolyte, by avoiding overfeed and it is commonly believed to clean the underside of the anode and to bring carbon dust to the surface of the melt. However, these benefits are not sufficient to outweigh the damaging effect of hothouse gasses (CF_4 and C_2F_6). Therefore the frequency of anode effects is also an important monitored parameter of cell status. In modern cells, the number of anode effects per day is from 0.01 to 1. Low anode effect frequency is preferred in order to reduce the emission of hothouse gasses.

Alumina feeding

Alumina feeding method affects many important parameters of cell operation. A typical automated break and feed system comprises a pneumatically operated crust breaker beam and an ore bin capable of discharging a fixed volume of alumina each time its gates are opened. The amount of alumina added in each dump, and the time interval between the dumps are usually constant. However, the time period between two dumps may also be varied, according to different programs or a demand feed signal. Small amount of alumina additions are considered advantageous in order to reduce the formation of alumina sludge on the bottom of the cell. Currently, the point feeding has become an important feature of modern prebake cells. The breaking of crust at two to four positions along the centerline of the cell, and the frequent but small alumina additions of the order of 1 kg per dump, make overfeeding less likely to occur. More frequently controlled and repeatable alumina additions can better stabilize the cell operation.

Cathode

Cathode failure will restrict the life of the cells and affect the quality of aluminum. Such failure is mainly due to bad operating strategies and cathode construction materials. Normally the operator cannot closely monitor the cathode failure. But different signals are of help to determine such failure, such as the iron impurity content in the metal, the voltage drop across the cathode lining and the observation of excessive deformation of the steel casing. All of these phenomena are utilized for relevant failure diagnosis. Among them, the worse occurrence would be to get a tap-out of metal directly through the bottom of the cell or through the bottom along the collector bars that may dissolve the steel shell. This type of failure causes the cell to be shut down.

Cell voltage instability

In operating cells, the cell voltage is an important control parameter, as the variation of cell voltage characterizes the instability status of a cell. Cell voltage is influenced by many factors, such as: alumina concentration, solvent electrolyte composition, cell temperature, amount of sludge in the cell, depth of the metal pad and the anode to cathode distance (ACD).

Anode

The normal life of the prebake anodes is about three weeks. Factors affecting anode consumption include cell operation, anode properties, current density, operating temperature. In the failure diagnosis process, the following observations reveal the occurrence of anode failure: color of the flame and anode stem, air burning above the anode and gases escaping through the crust holes, anode top covered with alumina and anode carbon particles floating on the electrolyte.

Fluoride addition

The cryolite is the best solvent for alumina. Various additions to the cryolite modify its physical and chemical properties and thus improve the cell performance. The most important additives used commercially are fluoride (2-10 mass%) and calcium fluoride (up to 8 mass%) both of which lower the freezing point of the electrolyte. Because there are advantages and disadvantages of the various additives, no such thing exists as an optimum universal electrolyte. Each cell design and operating process defines the precise requirements. Normally, the smaller cells that were fed infrequently used fairly similar electrolytes, these containing a limited excess of aluminum fluoride with calcium fluoride as the only other additive.

Sludge

The major causes of sludge are related to the operating problems or mechanical problems in the feeding system. The latter involves a breaking of the frozen overlaying crust and alumina feeding. An excessive amount of sludge will change the anode effect frequency and will also destabilize the metal pad. Because sludge is a poor electric conductor, its presence adversely affects cell performance. Therefore, it is necessary to prevent any great accumulation; otherwise a “sick pot” will result. The sludge problems are diagnosed based on the following symptoms: higher cathode voltage drop, irregular cathode current distribution and reduced anode effect frequency.

Tapping

For stable cell operation, it is important that the amount of metal removed balances the production in the time interval. The amount of tapped metal at each time is an important factor to judge the relevant fault. Because of the temperature of metal and bath, the distance between the anode and cathode will be changed depending on the amount of tapped metal. Therefore, in order to retain optimum operating conditions the metal produced is removed at regular intervals, and thus avoids major disturbances to the cell operation. Also, the amount of metal during a daily operation should be small enough to prevent significant alteration in heat distribution. The size of the cell or the quantity of tapped metal must be considered during the relevant diagnosis process.

2.4 Cell construction, Söderberg and prebake cell design

To design an expert system for the real process, the production technology is another fundamental consideration. Different types of cell and relevant production technologies will lead to different system designs.

There are two technologies applied in present smelters: Söderberg and prebake.

2.4.1 Söderberg technology

The continuous self-baking Söderberg anode utilizes heat generated within the cell to bake the anode paste, which is filled at the top of the anode. The paste is slowly baked as the anode moves downwards as a consequence of anode consumption. Steel studs bring the current to the anode. There are two types of Söderberg anodes, "Vertical Stud Söderberg" (VSS) and "Horizontal Stud Söderberg" (HSS). VSS cells are more common as they are less labor-intensive and cause fewer operating disturbances. Figure 2.2 shows the general structure of a VSS cell.

The two most important advantages of Söderberg anodes compared to the prebake anodes are a substantially lower plant capital investment (as the prebake anode forming and baking plants are eliminated), and the fact that continuity of the anode minimizes operating disturbances. But, the disadvantage of Söderberg anodes is their inferior anode carbon quality, which leads to a higher anode consumption. They also cause more negative environmental impacts, and require more complicated emission cleaning systems [6]. While significant progress has been made in improving its environmental performance, Söderberg technology is gradually being replaced with prebake technology. Currently, only about 25% smelters still use Söderberg technology.

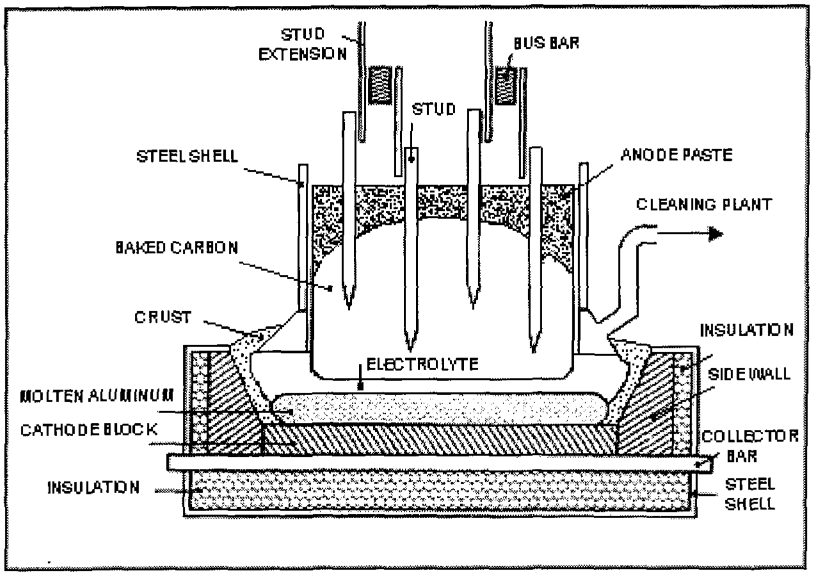

Figure 2.2 Vertical stud Söderberg cell

For routine operation, several aspects should be of concern to our expert system diagnosis process:

- Due to the specific forming process of the Söderberg anode, the carbon dust formed thereby is a major factor in excessive anode consumption and irregular pot operation.
- Due to the structure of the Söderberg anode cell, alumina is fed in large doses at the sides of the cell every few hours. This feeding method, with the slower dissolution of alumina and crust, will lead to sludge formation.

2.4.2 Prebake technology

Prebake technology uses multiple anodes in each cell, which are prebaked in a separate facility and attached to rods that suspend the anodes in the cell. When an anode is consumed, it is removed and replaced by a new one. The general structure of a prebake anode cell is shown in Figure 2.1. Prebake anodes are made from a blend of petroleum coke and coal tar pitch binder and baked in furnaces before being introduced to the cell. The baking process should produce cake from the binder of a quality approaching that of the aggregate material. This will minimize preferential oxidation of the binder cake and reduce net carbon consumption and anode dusting in cells. The prebake technology has some advantages compared to the Söderberg technology:

- Better compaction and quality control of prebake anodes can be maintained, which leads to a lowering in carbon consumption as well as making the cell control easier.
- The break and feed system keeps the amount of alumina additions and feeding frequency under control and that will keep the concentration of alumina more constant and will reduce the frequency of anode effects. It also can reduce the formation of sludge and can lead to more stable cell operation.
- Efficient current conductors and distributors between anode and anode studs, these lead to the low voltage drop.
- Other significant benefits are reduced emission of pot gases and improved operational efficiencies.

On the other hand, there are some disadvantages associated with prebake technology, which must be considered in the diagnosis process of the expert system:

- Point feeding leads to alumina being less preheated, and the frequent crust breaking can cause increased heat losses from the cell. The sludge, which in the case of point feeding is formed at center part of cell, is difficult to disperse.
- The anode's service life is about three weeks; therefore the anode changing disturbs the stability of cell operation, while the crust breaking also increases the heat losses.

2.4.3 Performance comparison between Söderberg cell and prebake cell

The main operation parameters of modern Söderberg and prebake cells are given in Table 2.1 **[2]**:

Table 2.1 Main performance parameters of Söderberg and prebake cell

Parameter	Söderberg cell	Prebake cell
Line current (kA)	125	280
Anode current density (A/cm^2)	0.78	0.72
Current efficiency (η)	92.5	96
Anode effects per cell day	0.4	0.1
Daily production (kg)	930	2146
Energy consumption (kWh/kg)	16.3	13.3

CHAPTER 3 FUNDAMENTALS OF EXPERT SYSTEM

3.1 Introduction to artificial intelligence

Artificial intelligence (AI) is a field of study that combines science and engineering in order to build machines capable of intelligent behavior. AI as a science tries to understand human intelligence, the nature of knowledge and the thinking process. Building an intelligent computer system requires us to understand how humans capture, organize, and manipulate knowledge during their problem solving. AI as engineering tries to build intelligent machines, such as robots and intelligent banking machines. In order to build these intelligent machines, AI uses many techniques. For example, expert systems use the specialist knowledge that people such as doctors, lawyers, industrial process operators and engineers have in order to assist other people. Machine learning is the study of how computers and robots can learn from their experiences. Neural networks are computer algorithms that work similarly to brains. Natural language processing studies human languages, like English or Japanese, by trying to teach computers to understand them **[1]**.

3.2 What is an expert system?

3.2.1 Definition of an expert system

An expert system is a computer program that reasons in a narrow but deep field of expertise. It emulates the decision-making ability of a human expert and will perform as well as -- if not better than -- humans operating in the same field. An expert system can manipulate the knowledge as well as the data. It can be used to represent human knowledge in a particular domain and then use a reasoning mechanism to manipulate this knowledge to provide advice. Expert systems differ from the conventional application programs in that:

- The main function of conventional programs is to store and to retrieve data, and to carry out calculations and to do graphics. A conventional program cannot reason with the knowledge. On the other hand, an expert system stores and retrieves knowledge and reasons with it.
- Expert systems simulate human reasoning about a problem in a narrow domain. They focus on emulating an expert's problem solving abilities.
- Expert systems solve problems by heuristic or approximate methods, which unlike algorithmic solutions, are not guaranteed to succeed. Such methods do not require perfect data and the solutions derived by the system may be proposed with varying degrees of certainty.
- Expert systems are capable of explaining and justifying solutions or recommendations, which helps the user to judge if the reasoning is in fact correct **[2]**.

The expert systems represent the expertise as data or rules that can be called upon when needed to solve problems. Books and manuals have a tremendous amount of knowledge but a human has to read and to interpret the knowledge for it to be used. However, the conventional program can use conventional decision-making logic that contains little knowledge for solving some specific problems. This programmed knowledge is often embedded as part of the programming code, so that as the knowledge changes, the program has to be changed and then rebuilt. But the knowledge-based systems can collect the small fragments of human know-how into a knowledge base, which is used to reason through a problem, using the knowledge that is appropriate. A different problem, within the domain of the knowledge base, can be solved using the same program without reprogramming. The ability of these systems to explain the reasoning process through back-tracking and to handle levels of confidence and uncertainty provides an additional feature that conventional programming does not handle. In summary, expert systems encode the domain dependent knowledge of experts in some field, and use this knowledge to solve problems **[3, 4]**.

3.2.2 Fundamental features of an expert system

To comprehend an expert system, and further to construct an application of an expert system, the following fundamental features should be considered at the beginning **[1]**.

<u>Expert systems applications</u>

Generally, expert systems can be applied in two different ways:

- Decision support: Providing information or options to an experienced decision maker. Commonly used in medicine.
- Decision-making: Allowing an unqualified person to make a decision beyond his or her level of training or expertise. Commonly used in industrial systems.

Ability of expert systems

- Perform at level of a human
- Recognize problems
- Recognize solutions
- Explain the proposed solution
- Select applicable solutions
- Deal with incomplete information
- Restructure problems
- Reduce the need for research
- Solve simple problems easily
- Sometimes can explain their reasoning
- Occasionally make judgments about the reliability of their conclusions
- Can build on existing knowledge.

Properties of expert systems

- Ask appropriate questions (based on external stimuli such as sight or sound)
- Reformulate questions to obtain answers
- Explain why they asked the question
- Explain why conclusion reached
- Judge the reliability of the conclusions
- Communicate easily with other experts in their field
- Reason on many levels and use a variety of tools such as heuristics, mathematical models and detailed simulations
- Transfer knowledge from one domain to another
- Use their knowledge efficiently.

We can see that there is quite a marked difference between the abilities of a human expert and a machine expert through the characteristics listed above. These properties are remarkably different from properties of a conventional program. Conventional programs basically depend upon the accuracy and integrity of the models. Therefore, if any of the input data is missing or inaccurate, the conventional system will respond with error messages or it may output incomprehensible results. Whereas an expert system can operate in the face of adversity, it does not need all the data to be accurate; it can use its reasoning facility to fill in or circumvent the gaps and it will return with results that include an estimate of reliability.

3.2.3 Historic background

The beginning of AI can be seen in the first game playing and puzzle-solving programs at the end of the 1940s. The fundamental idea of early research is called "state space search." The simplest form of state space search is "generate-and-test." There are two main variants of basic generate-and-test: "depth-first search" and "breadth-first search." Another important search algorithm developed is "heuristic search," which uses one or more items of domain-specific knowledge to traverse a state space graph. A heuristic is best thought of as a rule of thumb. Although not guaranteed to work in the decision procedure, a heuristic is useful in a majority of cases.

Beginning in the mid-1960s, the first goal in developing expert systems was to make machines "understand" natural language, especially stories and dialogue [5]. Other attempts in the same period were aimed at modeling human problem-solving behavior on simple tasks, such as puzzles, word games and memory tests. This idea was to simulate the use of knowledge and strategy the same way that a human would. This was exploration of goal-directed reasoning [6].

From the latter half of the 1970s to the present day, expert system researchers worked on the development of techniques and applications. This period is characterized by an increasing self-consciousness and self-criticism. The development of the problem-solving method, such as heuristic search, continued. Researchers developed techniques for encoding human knowledge in modules, which can be activated by patterns. These patterns may represent raw or processed data, problem states or partial problem solutions [7], [8].

With the emergence of computer technology, the expert system technology entered its implementation period, which was the first step towards applied expert systems. During this period, a variety of individual expert systems were completed, covering various application areas. In the 1970s, one of the first expert systems, MYCIN, was developed at Stanford University. MYCIN uses rules (coded in the computer language LISP) that contain medical knowledge to perform medical diagnoses.

During the early stages of development, expert systems were exclusively based on expert knowledge. Today, the term expert system may refer to any system that uses expert system technology and may not necessarily be based on expert knowledge. Also today, we use the terms expert system, knowledge-based system, and knowledge-based expert system synonymously **[2, 9]**.

3.2.4 Composition of an expert system

3.2.4.1 Expert system structure

An expert system is composed of three main components: the knowledge base (or rule base), the working memory and the inference engine. Figure 3.1 shows a block diagram of the elements of an expert system. The knowledge base is the source of facts, the working memory stores the processing data or inferred facts and conclusions, and the inference engine is used to draw conclusions or expertise about the user's query.

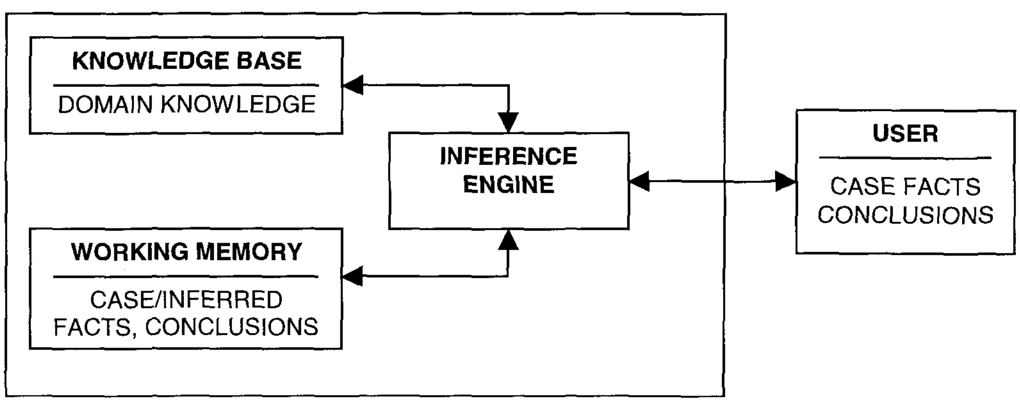

Figure 3.1 General structure of an expert system [1]

3.2.4.2 Elements of expert system

Knowledge base

An expert system maintains the expert's domain knowledge in a module called the knowledge base. Such knowledge acquired from the experts is coded by using several techniques. One of the typical ways of presenting knowledge in an expert system is a rule that is an IF/THEN structure that logically relates information contained in the IF part to other relevant information contained in the THEN part. The knowledge base consists of at least two types of data:

- Data that describes the problem and includes information that has been concluded, assumed or inferred.
- Knowledge that describes how to use the assertion base

Working memory

The working memory contains the facts (or information) about a problem that are used during a consultation. The system matches this information with knowledge contained in the knowledge base to infer new facts, then enters these new facts into the working memory, and the matching process continues. Eventually the system reaches a conclusion, which also enters into the working memory. The working memory will load the information contained in external storage such as databases, spreadsheets, or sensors at the beginning of the consultation process. Sometimes, the system may obtain the information supplied by the user. However, the working memory contains all the information about the problem that is either supplied by the user or inferred by the system.

Inference engine

The inference engine is a processor in an expert system that matches the facts contained in the memory with the domain knowledge contained in the knowledge base to draw conclusions about the problem. When an expert system is started to examine the problem, it searches the rules for a match between the premise and the information contained in the working memory. When the inference engine finds a match, it adds the rule's conclusion to the working memory and continues to scan the rules looking for new matches.

Either of the following two types of control strategies can be used in an inference engine:

- Forward chaining starts with assertions about the problem, makes inferences and draws conclusions. This strategy is used when all the knowledge to make a decision is available before session begins.
- Backward chaining or (goal driven) starts with the answer and works backwards to the problem description. The rule selection is guided by the conclusions rather than the conditions. This strategy is used in situations where the user can make a good guess about a possible solution and when more goals exist than combinations of initial assertions.

User Interface

The interactive communication between the expert system and the user is conducted in natural language, graphics or menus. The user interface is the means by which the user gains access to the expert system. If the system requires more data to solve a problem it will ask the user questions during a consultation. The expert system will communicate the results of a session to the user and will provide details of how its conclusion was reached and information about the reliability of the conclusion. A basic design requirement of the interface is to ask questions to obtain reliable information from the user **[2, 10, 11]**.

3.2.4.3 Expert system programming tools

There are two main expert systems building tools, programming language and expert system shells. The choice of a tool will depend on problems to be solved.

3.2.4.3.1 Programming language

An expert system can be programmed in either a traditional language or in an AI language. The traditional languages such as Fortran, Pascal, C and C_{++} are general-purpose languages and they have very good support facilities. But the AI languages such as LISP and PROLOG are given a narrower field of use. In comparison to traditional languages, the AI languages have the following features:

- Greater flexibility to design the knowledge base, inference engine, user interface and explanation program
- Greater skills of programmers and more time are required to learn the language

3.2.4.3.2 Expert system shell

Most expert systems are developed with specialized software tools called expert system shells. An expert system shell is a software development environment containing the basic components of expert systems and the methods to build them.

Brief features

An expert system shell can be seen as an expert system with an empty knowledge base. In principle, the knowledge in a particular domain can then be stored in the empty knowledge base and the system can then reason with this knowledge. Therefore, a shell is a domain-independent knowledge-based expert system. A shell can also have all components required for the knowledge base construction: Inference engine with different searching mechanisms; explanation facility and tools for writing hypertext, for constructing friendly user interfaces; and a knowledge base management facility. Hence, to build the expert system all that is required is that the knowledge base be constructed. The generic components of a shell are shown in Figure 3.2.

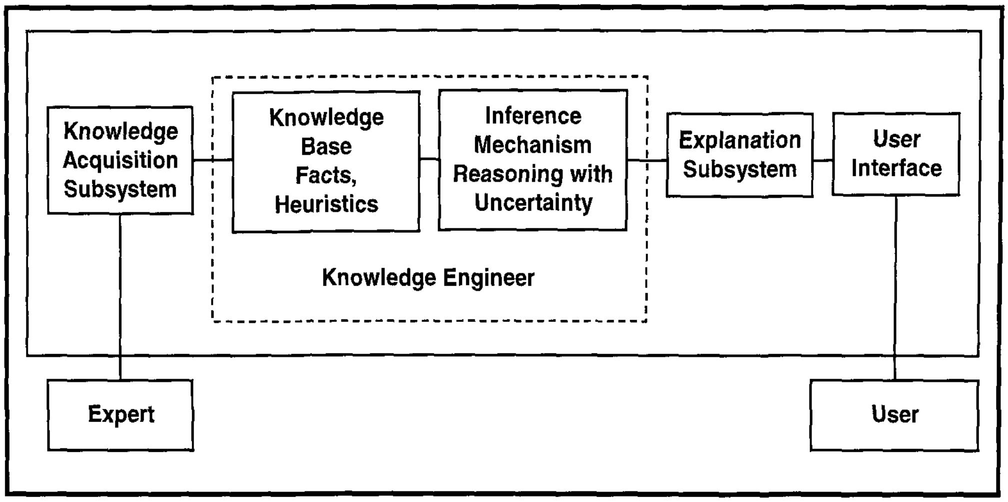

Figure 3.2 Generic components of an expert system shell [12]

Core components

As mentioned above, an expert system shell can be seen as an expert system with an empty knowledge base. Therefore, most components of the shell are similar to those of an expert system, except there are two additional components, knowledge acquisition sub-system and explanation sub-system, which are used for system developing. The definitions of them are:

- Knowledge acquisition sub-system: A sub-system to help experts build knowledge bases.
- Explanation sub-system: A sub-system that explains the system's actions. The explanation can range from how the final or intermediate solutions were arrived at to justifying the need for additional data.

However, they are not without limitations. Most expert system shells are developed with the inference engine having a specific reasoning mechanism and with a knowledge base having a specific knowledge representation language. Therefore, a particular shell might not be suitable for all types of problems. For example, certain shells are best for diagnosing problems while others are best suited for planning. For serious diagnostic applications, a shell would require some customization in order to deal with the problem.

Depending on the features of the shell, the differences between shell and AI language can be summarized as follows:

- Advantages of using shells
  - Easier to use
  - Generally require little programming knowledge
  - An expert system can be built easily and quickly.
- Disadvantages of using shells
  - Inflexible: Some shells may not fit non standard problems
  - Some shells can deal with one type of problem
  - Often costly to purchase.

There is no clear-cut answer on when to choose an expert system shell or a programming language. The choice depends on a number of things **[11]**:

- The application
- The development time available
- The skill level of the knowledge engineer
- The availability of the appropriate tool
- The budget available
- The level of support available.

In the present work Comdale and Gensym G2 shells were chosen. There are many other commercial tools in the market, such as EXSYS, ACQUIRE, Intellix, Expert Optimizer, EZ-Xpert, XpertRule, and CLIPS. They are broadly applied for general, task, or solution specific purposes.

3.3 How does an expert system work?

3.3.1 Knowledge engineering

Knowledge engineering is the process of developing a knowledge-based system by capturing, encoding, and testing domain-specific knowledge. There are six steps to be done for system building and maintenance:

- Problem selection
  The first step in knowledge engineering is selecting the "right problem", which is the goal of the project.
- Knowledge acquisition
  The objective of knowledge acquisition step is to acquire the knowledge of the problem, which is the foundation of expert system development.
- Knowledge representation
  This step involves representing the knowledge in the knowledge base as rules, frame scripts, semantic networks, or some combination of them.
- Knowledge encoding
  This step entails using the expert system shell/programming language to encode the knowledge.
- Knowledge testing and evaluation
  The major task of this step is to validate the overall structure of the system and its knowledge.
- Implementation and maintenance
  To periodically refine or update the knowledge to meet current needs after the system has been implemented.

3.3.2 Knowledge representation

Knowledge representation is a method used to encode knowledge in an expert system's knowledge base. To do this, the knowledge is represented in a symbolic form that can be manipulated by the expert system. There is no single knowledge representation system that is optimal for all applications. The success of the expert system depends on choosing the most appropriate knowledge representation for the problem to be solved.

3.3.2.1 Types of knowledge representation techniques

The following are five of the most common techniques used in an expert system to represent the knowledge:

- Object-attribute-value triplets (O-A-V)
- Rules
- Semantic networks
- Frames
- Logic

The first two are used in this thesis. Compared to other techniques, simple descriptions are used for structuring knowledge and controlling the reasoning process.

Object-Attribute-Value Triplets

O-A-V triplet structure can be used for complex propositions. It divides a given statement into three distinct parts: object, attribute, and attribute's value. Consider for example the statement, "The color of the anode stem is red." We can represent this statement in an O-A-V structure by defining the object as "The anode stem," the attribute as "colour" and the value as "red." The object represented in an O-A-V can be a physical or an abstract item. The attribute is a property or feature of the object. The value specifies the attribute's assignment. Also, one object can have multiple attributes, and each attribute also can have multiple values. Figure 3.3 shows an example of multi-attribute, multi-value O-A-V structure.

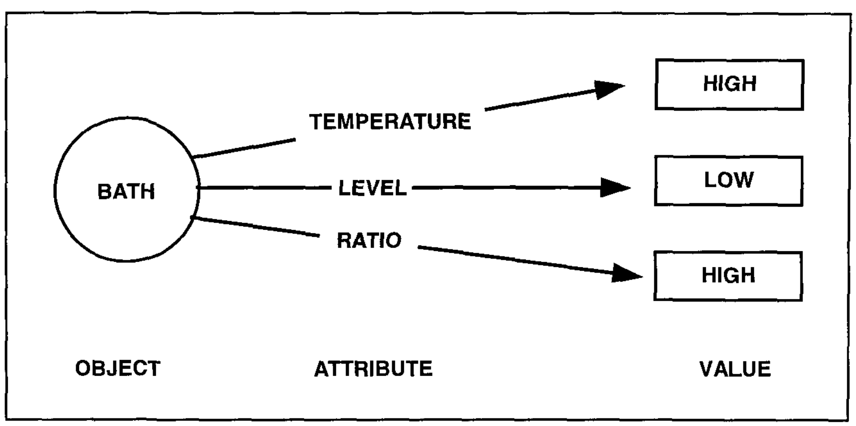

Figure 3.3 Example of multi-attribute and multi-value O-A-V triple structure

Rules

A rule is a knowledge structure that relates some known information to other information that can be concluded from known. A rule is a form of procedural knowledge that associates given information to some action. Rules represent reasoning knowledge and handle the complex relationship between facts. Rules can embody vague concepts, simple heuristics, mathematical expressions, date expressions, time expressions, character string expressions or functions. Rules are written in an IF-THEN-ELSE format. Whereas the IF-AND-OR part of the rule is called the "premise", the THEN-ELSE part is called the "conclusion". An example of a rule is:

> ***IF*** *anode effect number* ***is*** *high*
> ***AND*** *crust hardness* ***is*** *very high*
> ***THEN*** *bath temperature* ***is*** *low*
> ***THEN*** *text ("The bath temperature is low! Please check the line current source (rectifier, power station, etc.) and increase cell voltage temporarily".)*

In rule-based expert systems, domain knowledge is coded as a set of rules and entered in the system knowledge base. The system uses these rules along with information contained in the working memory to solve a problem. When the "IF" portion of the rule matches the information contained in the working memory, the system performs the action specified in the "THEN" part of the rule. The inference engine manages all these processes. To perform more complex operations, most rule-based systems are designed to access an external program such as common traditional software like a database or C program. As some facts have uncertain properties, the uncertain rules will provide an ability to establish an inexact association between the premise and the conclusion.

Rules have been widely used in many expert system applications because of the following advantages:

- Easy to understand, to modify and to maintain
- Inference and explanation easily produced
- Uncertainty is easily combined with rules
- Good for procedural knowledge. Suits wide range of heuristic knowledge.

Compared to other knowledge representation methods, some disadvantages of using rules are:

- Complex knowledge requires many rules
- Search limitations when there are many rules.

Semantic Networks

Semantic networks are the earliest attempt in AI to represent knowledge by computer programming. In semantic networks, pieces of knowledge are clustered together into relevant semantic groups, usually in the form of a graphical representation of knowledge showing relationships between objects **[11]**.

Frames

The frames differ with rules and semantic networks; in this technique, knowledge is decomposed into highly modular pieces called frames, which are generalized record structures. A frame is a package data structure that contains typical knowledge about some concept or object and includes both declarative and procedural knowledge **[13]**.

Logic statements

Logic is the oldest form of knowledge representation in a computer. The logical formalism is a powerful way of deriving new knowledge from old as it can be used to conclude that a certain proposition is true or false based on the truth of already known facts. This can be extended to derive answers to questions and solutions to problems. Two of the most often used logics are propositional logic and predicate calculus (also called as predicate logic). The latter is the extension of propositional logic. The one of the key AI programming languages PROLOG (PROgramming in LOGic) is based on the predicate calculus. Propositional logic represents and reasons with propositions, statements that are either true or false **[4]**, **[14]**.

3.3.2.2 Reasoning process – problem solving (Forward and backward chaining)

Reasoning is a process of working with knowledge, facts and problem solving strategies to draw conclusions. This process is similar to that used by humans to solve problems.

Inference, the core technique of reasoning in an expert system, is used to derive new information from known information. An expert system performs inference using a module called the inference engine. The most common inference strategy applied in expert systems is known as modus ponens **[1]**. If the premise of a rule is true, then its conclusion is also true. Like the knowledge base, the inference engine contains rules, and facts that pertain to more general control and search strategy are applied by the expert system in the development of a solution **[15]**. The basic characteristics of inference engine are:

- Differentiate between knowledge that is relevant and that which is irrelevant to the specific problem
- Consider relevant knowledge in a logical sequence rather than with browsing or jumping to conclusions.

As a knowledge base of an expert system is usually very large, it is necessary to have inferencing mechanisms that search through the database and deduce results in an organized manner. In managing a consultation session, the inference engine will look at a strategy to find out how and when the session is to end. Depending on the finish strategy criteria, the inference engine uses its start strategy to begin the inference process. Then the corresponding inference strategy is applied to guide with the reasoning techniques of the inference engine. The major techniques of reasoning strategies fall into two broad categories:

- Forward reasoning (Data-driven)
    - Existing facts matched to rule antecedents
    - Matching rules result in consequences: conclusions.
- Backward reasoning (Goal-driven)
    - Select goal or conclusion; match to rule-consequences
    - Check for match between rule-antecedents and facts
    - Repeat until conclusion matches fact.

Forward reasoning is a search procedure or reasoning process using known facts to produce new facts and to reach a final conclusion. On the other hand, backward reasoning is a reasoning process, which starts with a desired goal and works backward, looking for facts and rules that support the desired result.

Forward reasoning

The reasoning process of problem solving naturally begins with information collection, then this information is reasoned with to infer logical conclusions. It is similar to a process that a doctor uses to make a diagnosis. This style of reasoning that uses a data-driven search, it is called forward reasoning (forward chaining). The inference strategy of forward chaining starts with a set of known facts, derives new facts using rules whose premises match the known facts, and continues this process until a goal is reached or until no further rules have premises that match the known or derived facts. The following example shows the process of forward reasoning **[16]**:

Consider a set of rules:

Rule 1: ***IF*** *A*	***AND***	*C*	***THEN*** *F*
Rule 2: ***IF*** *A*	***AND***	*E*	***THEN*** *G*
Rule 3: ***IF*** *B*			***THEN*** *E*
Rule 4: ***IF*** *G*			***THEN*** *D*

The strategy to prove that D is true, given A and B are true would be: Start with Rule 1 and continue on down until a rule that "fires" is found. In this case, Rule 3 is the only one that fires in the first iteration. At the end of the first iteration, it can be concluded that A, B and E are true. This information is used in the second iteration. This time, Rule 2 fires and adds the information that G is true. This extra information causes Rule 4 to fire, proving that D is true. This is the method of forward chaining, where one proceeds from a given situation toward a desired goal, adding new assertions along the way. In expert systems, this strategy is especially appropriate in situations where data are expensive to collect, but limited in quantity. Figure 3.4 shows a flowchart of simple forward reasoning process in a rule-based expert system **[15]**.

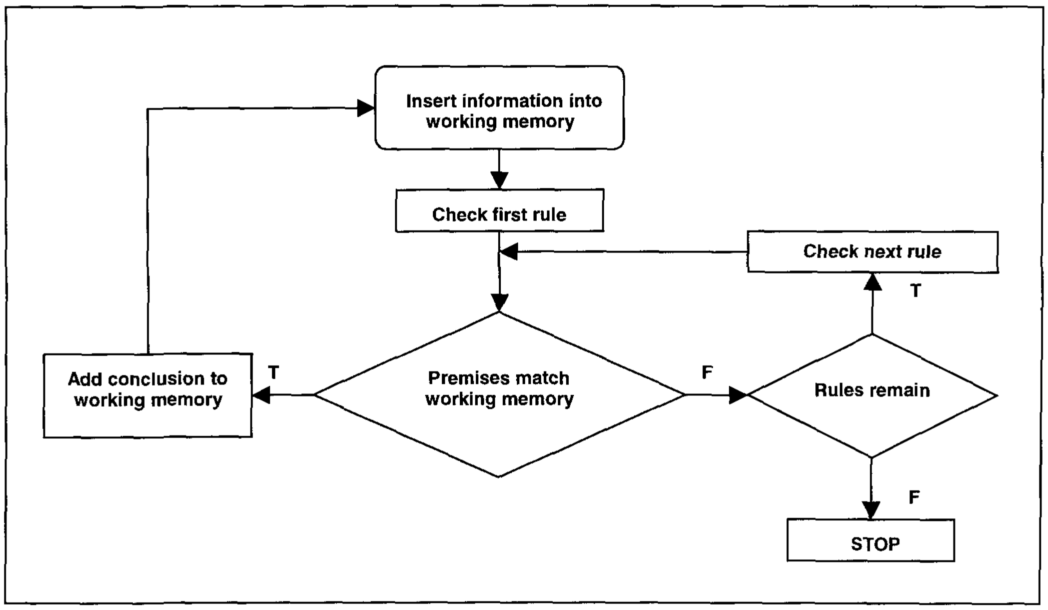

Figure 3.4 Forward reasoning process

Brief features of forward reasoning:

- Starts with the input data
- Examines data in a particular sequence
- Problem-solving mechanism keeps track of the implications of exampled fragments of the knowledge base each step of the way
- Continues until the implications are discovered to provide a solution to the problem.

Backward reasoning

Backward reasoning is another inference strategy that attempts to prove a hypothesis by gathering supporting information. This method starts with the desired goal, and then attempts to find evidence to prove the goal. First, backward reasoning checks the working memory to see if the goal has been previously added. If the goal has not been previously proven, the system searches its rules looking for one (or more) that contains the goal in its THEN part. This type of rule is called a goal rule. The system then checks to see if the goal rule's premises are listed in the working memory. Premises not listed then become new goals (also called sub goals) to prove that may be supported by other rules. This process continues in this recursive manner, until the system finds a premise that is not supported by any rule. Then the system uses this information to help prove both the sub goals and the original goal. The backward reasoning is similar to hypothesis testing in human problem solving.

As compared to the previous example of forward reasoning, the strategy to prove that D is true would be as follows. First, find a rule that proves D. Rule 4 does so. This provides a sub goal to prove that G is true. Now Rule 2 comes into play, and since it is already known that A is true, the new sub-goal is to show that E is true. Here, Rule 3 provides the next sub-goal of proving that B is true. But the fact that B is true is one of the given assertions. Therefore, E is true, which implies that G is true, which in turn implies that D is true **[15, 16]**.

Backward chaining is useful in situations where the quantity of data is potentially very large and where some specific characteristic of the system under consideration is of interest. Typical situations are various problems of diagnosis, such as medical diagnosis or faultfinding in electrical equipment. The brief procedure of backward reasoning is **[15]**:

- Starts with the original problem statement (goal)
- Decomposes problem into sub-problems (which could be broken into further sub-problems and so forth)
- Sub-problems could be easily solved by simply looking up a relevant fact or assertion in the knowledge base
- By solving all (or even some of) the sub-problems, the problem itself could be easily solved.

Depth-first and breadth-first search algorithms

A knowledge base can usually be represented as a branching network. There are many branching search algorithms available in the existing expert systems. However, only the two basic approaches, depth-first search and breadth-first search, are considered in our expert system.

The depth-first search algorithm begins at a node that either represents the given data (forward chaining) or the desired goal (backward chaining). It then checks to see if the left-most (or first) node beneath the initial node (call this node A) is a terminal node (i.e., it is proven or a goal). If not, it establishes node A on a list of sub-goals outstanding. It then starts with node A and looks at the first node below it, and so on. If there are no more lower level nodes, and a terminal node has not been reached, it starts from the last node on the outstanding list and takes the next route of descent to the right.

Breadth-first search starts by expanding all the nodes one level below the first node and then systematically expands each of these nodes until a solution is reached or the tree is completely expanded. The obvious advantage to this process is that the breadth-first search finds the shortest path from the initial assertion to a solution. However, such a search in large solution spaces can lead to huge computational costs due to an explosion in the number of nodes at a low level in the branch.

There are other methods of making inferences that use a combination of two or more of the above techniques. Depending on the number of given facts and the number of plausible inferences, some of these methods may be better than others in terms of time, memory and cost of the solution path **[16]**.

Brief features of the two reasoning techniques

Advantages of forward reasoning are:

- It works well when the problem naturally begins by gathering information and then seeing what can be inferred from it
- It can provide a considerable amount of information from only a small amount of data
- It works well for certain types of problem solving tasks, such as planning, monitoring, control and interpretation.

Disadvantages of forward reasoning are:

- The system will ask all possible questions; even though it may only need to ask a few questions to arrive at a conclusion
- The system may also ask unrelated questions.

Advantages of backward reasoning:

- It works well when the problem naturally begins by forming a hypothesis and then seeing if it can be proven
- It remains focused on a given goal
- The system searches only that part of knowledge base that is relevant to the current problem
- It works well for certain types of problem solving tasks, such as diagnostics, prescription and debugging.

A disadvantage of backward reasoning:

- It will continue to follow a given line of reasoning even if it should discontinue and switch to a different one.

3.4 The methodology of expert system development

In the beginning stage of an expert system development, the following basic questions are always considered:

- Why are we developing this expert system?
- How to get the knowledge?
- How are we going to build it?
- What tools are available?

With these primary considerations, the remaining steps in the entire developing stages are as follows:

- Tool studying
- Knowledge acquisition
- Knowledge analysis
- System design
- System validation and improvement.

3.5 Knowledge — the foundation of expert systems

What is knowledge? Knowledge is an abstract term that attempts to capture an individual's understanding of a given subject [1]. Expert systems drive their power from the knowledge they contain. Knowledge is the heart of any expert system and it is the effective use of knowledge that makes the system's reasoning successful. Like conventional programming, there is no single "best" data structure for all computing purposes. For an expert system, different knowledge representation methods are used for different applications; no single knowledge representation structure is ideal. Therefore, to find the knowledge and corresponding presentation methods best suited for a given application is the primary objective of a knowledge engineer.

To develop the expert system in this thesis, we do not attempt to capture all of the expert's knowledge of the aluminum industry. We only target the expert's knowledge focused on a narrow subject: aluminum electrolysis process, which is the so-called domain knowledge.

For expert system building, there are three important aspects to consider:

1. Understanding of the subject area (knowledge acquisition)

2. Focusing on the subject area (knowledge editing)
3. Using the corresponding method to encode knowledge to the knowledge base (knowledge representation).

It has been recognized that the performance of an expert system is directly affected by the quality of knowledge coded in the knowledge base.

3.5.1 Levels of knowledge

Human knowledge can be represented at different levels, which depend on the degree that fundamental principles and causal relationships are taken into account. Generally, human knowledge is described in two levels:

- Shallow knowledge, which handles only surface level information that can be used for the specific situations
- Deep knowledge, which represents the internal and causal structure of a system and considers the interactions between its components.

Shallow knowledge is concerned only with the type of information that is needed to solve a particular type of problem while deep knowledge can be applied to different tasks and different situations. For example, some experiences obtained from process operators belong to shallow knowledge, which reflects a direct relationship between cause and effect. On the other hand, the knowledge of experts or engineers belongs to deep knowledge. Depending on the complexity of the real process, both levels of knowledge are needed in expert systems.

3.5.2 The components of knowledge

In any knowledge domain, many different sets of objects may be used for knowledge representation. In the aluminum electrolysis process, for instance, the objects may be the particular cell types, such as Söderberg cells or prebake cells, and open or hooded cells. Once we have settled on a set of objects to use for knowledge representation, it is necessary to define them and their interactions. These definitions are based on the elementary components of knowledge:

- Naming – Give the proper nouns to specific objects
- Describing – Describe the property of an object, usually performed by adjectives
- Organizing – Handle the objects in a variety of ways including categorization and possession
- Relating - Use transitive verbs and special nouns to describe relationship
- Constraining – Handle the conditions that define what descriptions of objects or patterns of relationships between objects are admissible.

For example, in our system, "anode", "cell voltage" are specific objects names. "Bath temperature is high," "Cell voltage is not stable" describe corresponding objects, where "high" and "not stable" are adjectives of the objects "bath temperature" and "cell voltage." "Aluminum has a limit value on iron impurity" indicates an attribute of object "aluminum impurity." To describe the relationship between the objects, we use sentences such as "The crust is broken by the point breaker" to distinguish the relation of "crust" and "point feeder." Constraint is often used in our system to define running conditions, for example: "Any anode whose voltage drop is less than 0.1 V, may not perform normally."

These components of knowledge are used in knowledge representation methods. However, each knowledge representation method deals with these components in its own way [4].

3.5.3 Knowledge acquisition

Generally, the raw material of the knowledge of our system is collected in two different ways: from the open sources or from the real process. Normally, the knowledge collected from the open literature is considered for the general purpose knowledge base building, whereas the particular knowledge will come directly from the plant, as will the real process data applied for the respective cells. We have collected both of them.

3.5.3.1 Stages of knowledge acquisition

To build our knowledge base, the knowledge acquisition has been executed in three stages:

1. Elementary and visual observation of the process
2. Direct expertise from domain experts
3. Systematic expertise from senior experts.

The first step is not only elementary but also provides necessary knowledge for us. This kind of knowledge acquisition starts at the very beginning of the system design. It consists of two steps:

- Study of relevant books, papers and video materials
- Plant visit.

After this first stage, an elementary knowledge of the process was obtained.

In the second stage, the knowledge was directly collected from the domain engineers and operators. This is a more detailed knowledge of the aluminum electrolysis. The characteristics of the knowledge collected at this stage are:

- Much more detailed description of the production process
- Specific case studies for certain events, e.g. the anode effect.

The third stage is more important for our knowledge base building. The systematic knowledge provided by the domain senior experts gives us the following benefits:

- Easier to present the problems
- Systematically organize knowledge base
- Easier to create the rules base
- Faster to get the diagnosis results.

3.5.3.2 Improvement of acquired knowledge

Figure 3.5 [1] illustrates five general steps of a knowledge base building. Although the tasks are shown in sequence, in practice there is considerable overlap in their execution. Based on the knowledge acquired, the preliminary design of the knowledge base will be done; then iterations are needed during the system development. After several steps, the system will be improved from one with limited ability to one that becomes more capable due to its improved knowledge and problem solving skills. However the expert system can improve the valuable solutions to real world problems, which is based on learning from the problems occurred during the process of design, evaluation and routine running.

For our system design, we investigated the general concepts of this particular domain and created a formal representation of the objects and relations in the domain. Then, we collected and analyzed several open sources about the problem diagnosis and correction of the aluminum electrolytic process. Based on these sources, we designed the general structure of the expert system.

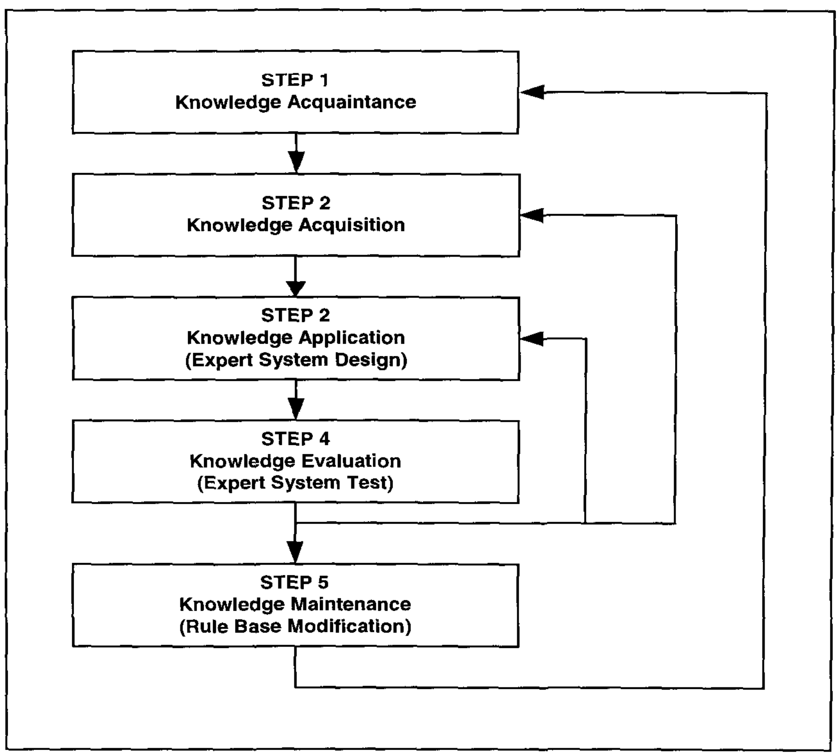

Figure 3.5 Steps of knowledge base building

The engineer level expert system was built from the generalized properties of the electrolytic process taken from the open sources we acquired. Then we built the operator level expert system, based on the expertise and data from a given particular type of cell.

To verify and improve the knowledge of the expert system, we asked experts of this domain to conduct a test run of our expert system. Their opinions were fed back to the knowledge base for further improvement.

3.6 Brief review on industrial applications of expert systems

After four decades of development, AI technology has progressively proved to be useful in industrial applications. Expert systems, artificial neural networks and fuzzy logic systems are three main types of AI applications. They are already frequently chosen as additional tools for the control engineer. For example, many large automation and process control companies have developed internal AI elements for their plant control system and fuzzy logic is already integrated in some chips, which are widely used in many areas [17,18].

Expert systems, one of the major branches of AI technologies, can be used in most processing applications, even very large ones needed for major chemical processes, metallurgical processes, quality control in pulp and paper and oil industries, cost control in power plants and other applications. Generally, operators and engineers use expert systems for fault diagnosis. Expert systems can find the most probable cause and suggest corrective actions. Expert systems also are very useful in alarm management because by communicating the critical alarms first, the operator can react immediately to correct the problem. This saves process downtime, operator time in locating the problem, and in the long run, saves money and reduces off-specification products. Real-time expert systems have also been developed for industrial processes, sometimes in conjunction with fuzzy logic. Using fuzzy logic sets, it determines set points and control actions. The expert system acts as a watchdog, which can increase throughput, lower operating costs, and increase yields. Successful examples of the implementation of expert systems in industry can be found in robot movement planning, chemical compound synthesis, computer systems configuration, decision support, fault diagnosis, and engineering design [19].

The major difference between expert system techniques versus classical algorithmic techniques is the ability of expert system techniques to develop systems that infer answers from complex and incomplete knowledge bases. In other words, just as an acceptable answer – although not necessarily the optimal answer – can be determined by humans, given incomplete and perhaps unrelated facts, an expert system can be made to choose rational and perhaps multiple answers from a set of rules and facts. Based on such abilities of expert systems, their applications for problem solving can be categorized as in Table 3.1:

Table 3.1 Expert system applications

Expert System Applications	
1	Control
2	Design
3	Diagnosis instruction
4	Interpretation
5	Monitoring
6	Planning
7	Prediction
8	Prescription
9	Selection
10	Simulation

In the reviewed literature **[20, 21, 22]** and the recent applications **[23, 24]**, *some interesting new characteristics of industrial applications are found, e. g:*

- Combinations of several AI technologies for one project
- Ease of use and installation
- Graphical/visual user interface
- Heuristic model based optimal control
- AI technique embedded in traditional process control system
- A higher level smart-alarm management (or alarm about alarms).

Although expert systems have many successes in industrial areas, very few have been actually installed in plants in the context of plant operation and control due to the following major reasons **[25, 26]**:

- Development is complicated; it is not simply selecting a shell, and obtaining a set of rules and transforming them into the syntax of the knowledge presentation language. In most cases, the rules are not initially available, and must be developed.
- Difficult to directly extend an application to the individual object due to the difference in their properties and corresponding knowledge acquisition.
- Knowledge acquisition is a difficult and lengthy process. As the expert system relies heavily on knowledge, the quality of knowledge often determines the success of an expert system. Knowledge acquisition is usually the "bottleneck" in expert system development.
- Expert system maintenance is difficult, since knowledge evolves continuously.

Recently some new developments in industrial applications of expert systems have appeared. In one, an expert system is used as a training tool in the context of computer-aided instruction. This type of application combines one expert system that provides domain knowledge with another expert system that has the know-how to present the domain knowledge in a learnable format. The system could then vary its presentation style to fit the needs of the individual learner. While this concept is not new, in combination with today's powerful personal computers, this kind of Intelligent Computer Assisted Instruction (ICAI) is much smarter than before.

Another important development concerns the expert system shells: advanced shells allow developers to embed inference engines into other kinds of programs. Embedded inference engines enable developers to set up a number of potential applications. Word processors are becoming intelligent. Smart spreadsheets can tell us the kinds of models we should use in our projections. All the components of expert systems can be easily embedded into any system **[27]**.

Additional documentation of expert system development and applications is given in references **[28 – 44]**.

CHAPTER 4 EXPERT SYSTEM APPLICATIONS FOR ALUMINUM ELECTROLYSIS PROCESS

4.1 Expert system applications review

Most applications of expert systems in aluminum production were developed experimentally at the end of 1980s and at the beginning of 1990s. Only a few articles on this subject have been published. To the author's knowledge, no expert system is presently used in a plant. The analyses of these publications give the ideas of what should be done and what should be improved in the present expert system. Now we continue with the presentation of these expert systems:

4.1.1 An integrated control-supervision system

L. Tikasz et al. described a proposal of expert system application for aluminum smelters **[1]**. This expert system is connected to the operator, the process control system and a process simulator.

Process control

In the traditional process control system, only the line current and the cell voltage are measured continuously, the other parameters related to the process are measured and logged occasionally. The expert system is suggested to compensate for this characteristic weakness of the present control system.

Process supervision

As the electrolysis is a complex electrochemical process, there are many interactions between the process and the operator. The proposed expert system would help to supervise the electrolysis cell process.

Process simulator

The process simulator consists of a lumped-parameter model and the cell control system. In the process supervision loop, the process simulator would be a predictive tool for variables that are not continuously measured.

The authors proposed the integration of these components, as shown in Figure 4.1, where, the real-time expert system plays a coordination and assistant role. The existing control and the operator are still the basis of the control. The simulator generates predictive data for the expert system and for the operator.

Analyzing the state of process, the expert system provides advice to the operator, who still makes the final decisions. The operator is in the center of the entire supervision system.

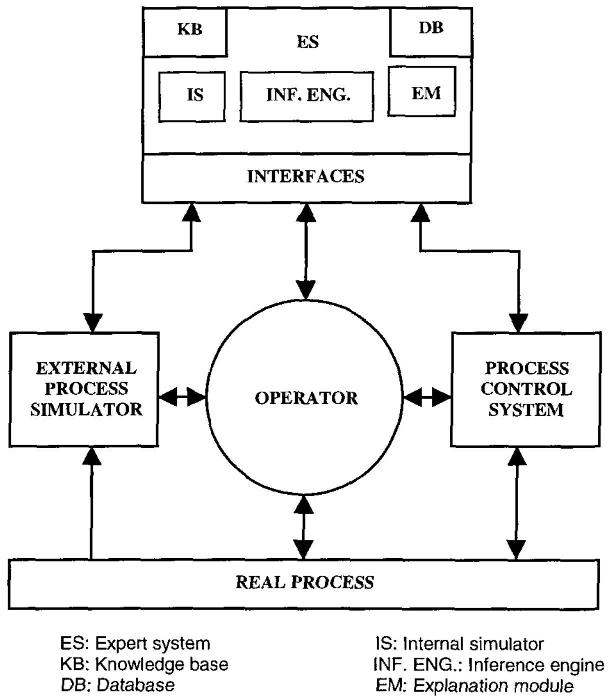

Figure 4.1 The integrated control-supervision system of Tikasz et al [1]

4.1.2 A hierarchical, intelligent control system

J. Li et al. developed a hierarchical, intelligent control system for aluminum reduction cells [2]. There are two intelligent levels in the system. The higher level is built on the principle of a neural network expert system aimed at analyzing medium and long-term change trends of the state of the process and calculating the set point for the lower level. The lower level is a fuzzy controller, which carries short-term analyses and real-time control of the process. The general structure of this hierarchical intelligent control system is shown in Figure 4.2.

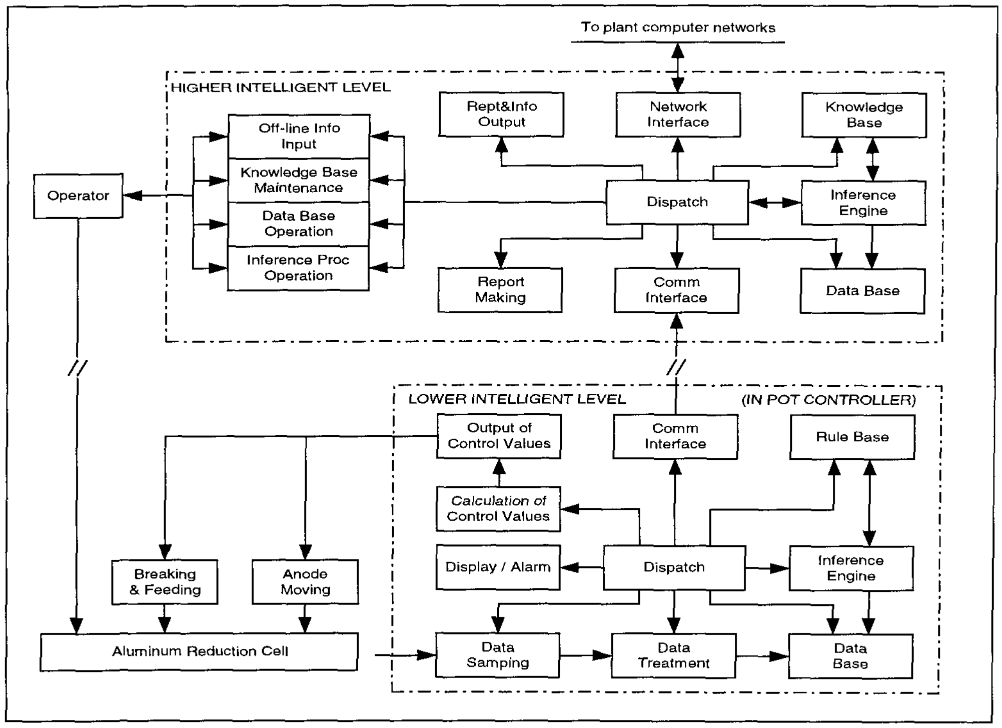

Figure 4.2 General structure of the hierarchical, intelligent control system of J. Li et al. [2]

Main characteristics of system

- In the higher level, the knowledge base is made of neural networks. Therefore the system is efficient in learning, extraction and parallel processing of knowledge.
- The higher level inference engine can start automatically or be activated by operators.
- The core of the lower intelligence level is an expert fuzzy controller, which combines fuzzy control techniques with expert system approaches.
- The lower level Inference engine uses the forward reasoning to obtain a conclusion.

This system has not been applied to a plant, but test runs have demonstrated potential for plant implementation.

4.1.3 A consulting toolkit

L. Tikasz et al. developed another expert system, which is a consulting toolkit for aluminum electrolysis **[3]**. This is an off-line, consultation-type expert system that focuses on constructing a modular consulting toolkit to help the process diagnosis and the situation assessment. It was developed with Comdale/X software package and can be used for the typical applications: diagnosis, personnel training and operator guidance.

Knowledge base development

The principle of setting up a knowledge base is available for reference:

- Search for domains where the knowledge is worthy.
- Draw some substantial experience from two previous expert systems. The first system was concerned about how to use an expert system for control. The second system carried out problem detection related to alumina supply to the cell.
- Knowledge was also collected in the fields of diagnosis and correction of irregularly operating cells, noise analysis and voltage balance measurement.

Brief features

Compared to other expert systems, some different approaches were adopted:

- Noise analysis
  The FFT (Fast Fourier Transform) and statistical methods were proposed for noise analysis, which includes the random noise and waving noise. The purpose of noise analysis was to demonstrate how a traditional manual could be converted into expert system applications.
- Mixed expert system
  A mixed approach was suggested by using the noise analysis results. In this case, the expert system environment or the Hypertext document is migrated according to the user's needs.

Strong points

- The system served as a good reference for our present knowledge base because the knowledge was based on plant experience.
- Noise is a main factor that affects the measurement accuracy both in a traditional control system and in an expert system. Noise treatment in the expert system improves its performance.
- For best results, the combination of a simulator and an expert system would be a useful training tool.

4.1.4 HALDRIS expert system

W. K. Rolland and his colleagues developed the HALDRIS (Hydro Aluminum DRIfts-Støtte) expert system, one of the few expert systems that have been applied in the smelter process **[4]**. This expert system was designed for assisting the operators of 125 kA VSS pots. HALDRIS was designed to keep potline on target. It helped the operators supervise and carry out corrective actions on individual cells. This expert system used the Gensym G2 expert system shell.

Knowledge representation and implementation

Most knowledge acquired came from a set of documented guidelines. These guidelines were updated first and then were implemented into the knowledge base. Some knowledge was also obtained directly from the operators.

Structure of HALDRIS system

The HALDRIS system runs on a dedicated UNIX workstation connected to an Ethernet. The process database, which receives data from the measurement system, as from manual inputs, is connected to the same network. The graphics and windows based interface helps user to understand the state of the cell quickly and effectively. The process role of HALDRIS is schematically shown in Figure 4.3.

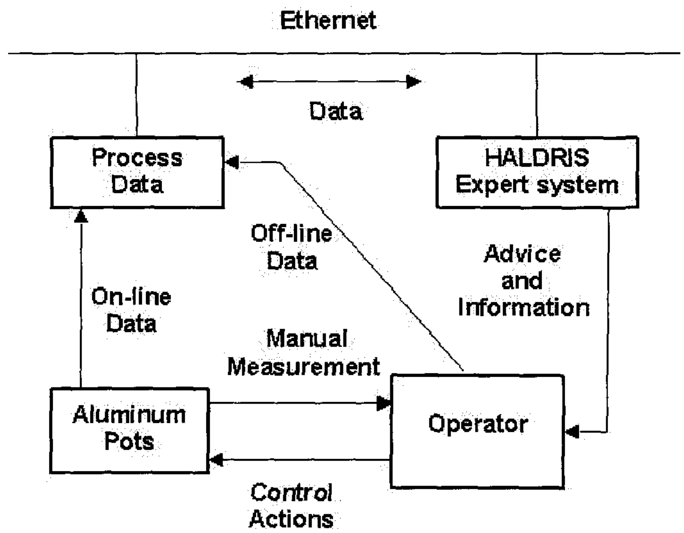

Figure 4.3 The process role of the basics of HALDRIS

Brief features

- Two main tasks of daily operation are:
    - To direct the operator's attention to pots that show abnormal behavior.
    - Based on the stated information of the focused cell, to allow the operator to obtain an explanation or comment on the information.
- The data can be updated from the database, which is managed by the expert system itself. However, only the most important process data are extracted from a large amount of measurement data.
- HALDRIS has a hierarchical main menu system. Desired options are selected with a mouse.
- The cell state displays three levels (See Table 4.1), that allow users to quickly understand the cell state:

Table 4.1 Cell state display list

Cell State Display	
1	Numerically
2	Symbolically
3	Visually

Strong points

- If HALDRIS gives a recommendation that an experienced operator does not agree with, the operator is given the option of commenting on the message in which the advice was given, so that the situation may be saved for later discussions with other expert operators. This ability is helpful for knowledge base maintenance.
- It is possible for the operator to modify the actions through an appropriate interaction with the expert system. Such modification is put in effect by the questions asked by HALDRIS. The operator's input depends on supplementary input data and his interpretation of the situation, but HALDRIS helps the operator draw an overall conclusion.

4.1.5 Combination of control and expert system

Z. J. Liu et al. developed a new type of expert system **[5]**, where the traditional process control and expert system strategies are combined for electrolysis process. First, they developed a correlation curve between apparent resistance of aluminum cell and alumina concentration through industrial test and research. Then, based on the facts, analysis of test data and experience from experts, a fuzzy control model for alumina concentration was set up, that realized feeding on demand for aluminum cells and considerable reduction of anode effect.

Control model structure

The control model consisted of two major parts: an alumina feeding model (See Table 4.2) and an anode position control model (See Table 4.3).

Table 4.2 Alumina feeding models

	Alumina Feeding Model
1	Manual feeding model
2	Timing feeding model
3	Special operation feeding model
4	Alumina concentration fuzzy control model
5	Alumina concentration trading control model
6	Anode prediction model

Table 4.3 Anode position control models

	Anode Position Control Model
1	Manual anode position control model
2	Automatic anode position control model (1) Anode position fuzzy control model (2) Resistance/anode position control model (3) Special operation anode position control model

Characteristics of control strategies

- Alumina concentration fuzzy control strategy is based on the relationship between resistance and alumina concentration, and expertise acquired from experts and operators. Alumina concentration can be controlled at 1.0%~3.5%.
- An anode position control model is not only based on routine operations, but also takes into account some abnormal operations, which are absorbed from the operator's experience.
- An anode effect prediction is also based on the alumina feeding fuzzy control strategy. The anode effect frequency can be reduced to 0.25 anode effect per day per cell.

4.1.6 An intelligent pot control

An "Intelligent Pot Control" is another interesting project that was released in 2000. It is part of the U.S. Department of Energy's "Industries of Future" program **[6]**. The first strategy is an expert system that will assist pot operation personnel to maintain optimum operating conditions for aluminum electrolysis cells. Artificial intelligence is the second strategy to be employed; it can enhance the cell control system. These control strategies will be based on software developed by Gensym Corporation. The major goals of this project are significant energy savings and lower emissions as even small gains in performance can give substantial cost improvements and environmental gains.

4.2 Analyses of the expert system applications - Why choose an expert system for aluminum electrolysis process?

Based on the reviews of these expert system applications, the further analysis involves many considerations, such as system structure layout, knowledge base building, data communication bridge selection and interface design. All of them are concerned with a basic topic: "Why choose an expert system for aluminum electrolysis process?" Namely, what and how the expert system can do to improve of present production. Considering that the expert system will cooperate with the existing process control system, to analyze the limitations of a conventional process control system and to understand its fundamental control strategies is the primary concern. Then, we can find out the limited availability of the present process control system and how an expert system can complement these disadvantages.

4.2.1 Fundamental control strategies analysis

Electrolysis process control systems have been developed both in hardware and software. The control strategies of present cell control systems are essentially based on the following principles:

- Energy balance
- Material balance

A good energy balance helps stabilize the bath temperature and the "freeze" formation. "Freeze" is that part of the electrolyte that solidifies along the relatively cold carbon cathode sidewall and thus helps protect the cell cavity against the highly corrosive bath.

A good material balance helps keep the alumina concentration at or near the optimal values. Too high alumina concentration may lead to the formation of "sludge," which is difficult to remove and can affect the current distribution in the cell. Too low alumina concentration may trigger an "anode effect." In that case, a gas layer will build up under the anode, which may affect the cell resistance and finally the cell voltage.

One of the general principles of control strategy determination is based on monitoring the cell pseudo resistance, which is calculated from the measured cell current and voltage. Based on this essential information, the control logic may change the anode to cathode distance (ACD) by adjusting the anode position or by modifying the alumina feed rate to keep the target value of resistance.

Another important part of cell control deals with alumina concentration. The concentration is related not only to the cell resistance but also to other cell parameters. The control of alumina concentration is done by analyzing cell pseudo resistance, primarily because the sensors for alumina concentration are not available and therefore, there is no direct feedback.

The alumina concentration and its interrelationships are affected by:

- The amount of alumina added in a break and feed cycle
- The time elapsed since feeding
- The amount of ledge of the frozen electrolyte
- The amount of sludge being formed during feeding.

4.2.2 Difficult task of present process control system

Several process variables of the electrolysis process are difficult to control by traditional process control approaches mainly due to the lack of necessary information. For example, to control the aluminum concentration at the target value, many process variables must be considered. But only two process variables, the cell voltage and potline amperage can be automatically measured. Most of the reminder information is provided by the operators. This additional information is concerned with two types of data, variables and parameters. The variables are measured, calculated or analyzed to characterize cell operation. The parameters are the controls that can be deliberately modified to obtain corrective effects on the potline operation. Process variables generally are obtained by the following methods:

- Operator manual measurement
- Operator visual observation
- Analyses result from laboratory.

Normally this additional information is sent by the operators through two functions:

- Through switches and push-buttons informing the automatic equipment of manual operation being carried out on the cell (e.g. metal tapping, anode changing)
- Through data loggers which automatically feed the system with cell measurements.

The particular environment of the cell operation and particular materials used cause problems in process variable measurements. For example, temperature is one of the most important process control parameters, but continuous measurements still have not proven to be technically or economically viable. The thermocouples wear out rapidly as continuous measurements in the bath are made **[7]**.

Due to this situation, the present operating practice is to regularly check the cell temperature by using a metal clad thermocouple. However, it is difficult to keep a uniform procedure (fixed position and depth). The same situation can be found in other process variable measurements, such as the position measurements of the anode and the sludge.

As a consequence, the existing process control strategies still lack a very important basic link, which is the continuous measurement of the major process variables from the operating cells.

4.2.3 Comparison of expert systems

Expert systems have been applied to automation areas for decades. The knowledge applied is extracted from human experts of a special field. Therefore they can perform a rather difficult task usually performed only by humans. Expert systems provide new capabilities and flexibility in applying control strategies. These systems offer the capability to capture, retain and utilize valuable process expertise that has accumulated over years of plant engineering and process operation. The main applications in the process and manufacturing area are:

- Fault diagnosis and repair advice
- Alarm analysis
- Condition monitoring
- Process monitoring and recovery advice
- Quality control
- Process control and optimization
- Simulation and prediction
- Configuration
- Planning and scheduling
- Physical measurements interpretation
- Design
- Supervision.

However, as mentioned above, process control strategies for electrolysis cells are still based on the cell voltage (or equivalent pseudo resistance). The cell pseudo resistance is compared to a predetermined set-point value, and if the deviation is larger than a certain limit, the control action will be to adjust the interpolar distance by raising or lowering the anodes. Undoubtedly, this typical process control is efficient for the normal case, but when something unusual happens, for example an anode breaks or cathode fails, the control logic cannot detect it. The rule-based expert system is a good means to remedy the weakness of the conventional process control system.

We have found that some existing efficient control strategies are just based on the logic of human experience in the real process control area. An interesting example can be found in the "Anode effect treatment control" program. This program is automatically started by a signal from the process: "the cell voltage is higher than 10 V". But the variation of the voltage is related to many reasons, such as: the distance between the anode and metal level, the condition of the anode, the condition of the bath, and the alumina concentration. This is why in the "Anode effect treatment" control program the trial-and-error strategy is based upon some rules, which came from operator experience or expert knowledge. Thus, the philosophy of expert systems has already been adopted in some existing control programs, although they are not called expert systems. But the ability of such an improved control strategy is still limited, as it cannot be applied for solving more complex problems.

This example gives us a clear message that depending on the capabilities and flexibility of the expert system, operator experience and expert knowledge can be applied to solve some problems for which the conventional control strategy is not adequate.

4.2.4 Using expert system to improve cell operation

To improve the performance of the electrolysis cell operation, the design of the process control system is concerned with several objectives. Increasing the energy efficiency is an important goal. Normally, the average energy efficiency of the reduction cell is only 40-50%. Because of the limited knowledge about the physical properties and limited process variables measurement, it is difficult to make an attempt to increase energy efficiency by a conventional process control strategy.

Here is another example. The theoretical carbon anode consumption is 0.333 kg C/kg Al, but the operational value is about 0.4-0.5 kg C/kg Al. The large difference is caused by several factors:

- Anode quality
- Cell design
- Cell operation.

The anode consumption is strongly dependent on the cell operation. Some relevant operational factors are given as:

- Anode temperature profile
- Current efficiency
- Electrolyte composition
- Anode setting routines
- Anode gas outlets
- Crust breaking pattern.

As was mentioned above, because the process data measurements are limited, it may be difficult to make a correct decision by regular control routines for complex cases. For example, when the symptoms show that the cell voltage is unstable and higher than normal and the pot meter is swinging more than usual, if the conventional feedback control is applied, the control action should be: "Start the motor to raise anode until the voltage fluctuation is down to normal." What should be done in next step? May be only to change the set point of the anode until the voltage change is lowered. This is a right way to solve the voltage instability, but may not suit the more complex cases. If the expert system is used, depending upon its knowledge base, several conditions and factors can be examined:

- Low anode
- Spiked or grounded anode
- Lumps of carbon in bath-under anode
- Anode set on objects (lump, cryolite, carbon, etc.)
- Metal inversion

Depending on the knowledge acquired in the knowledge base and required information, the inference engine starts to reason and check all possible cases and finally gives a diagnosis result. This will help the operator to find the corresponding actions to actually solve the problem, and not only to modify the set point of the anode, which would be the limitation of the conventional control practice.

By studying the fundamentals of the aluminum electrolysis process, it is clear that when compared to the conventional process control system, the expert system can be a very useful tool for operational routines. However, depending on the capability and flexibility of the expert system, the expert rule or human experience can be applied not only for the expert system but also in combination with a conventional process control strategy to form a new type of advanced control system. Figure 4.4 shows the combination of an expert system and a conventional process control system, which applied in the present work. The expert system receives the relevant process information and does the reasoning. Then, the diagnosis result will advise the operators why the irregular cases occurred and how to solve such problems. Further than that, the diagnosis results and suggestions will be sent to the control system as reference value or conditions. For example, the diagnosis result "Anode position is too low" can be compiled into the conventional control program as an executive condition of the action of "Start to raise anode."

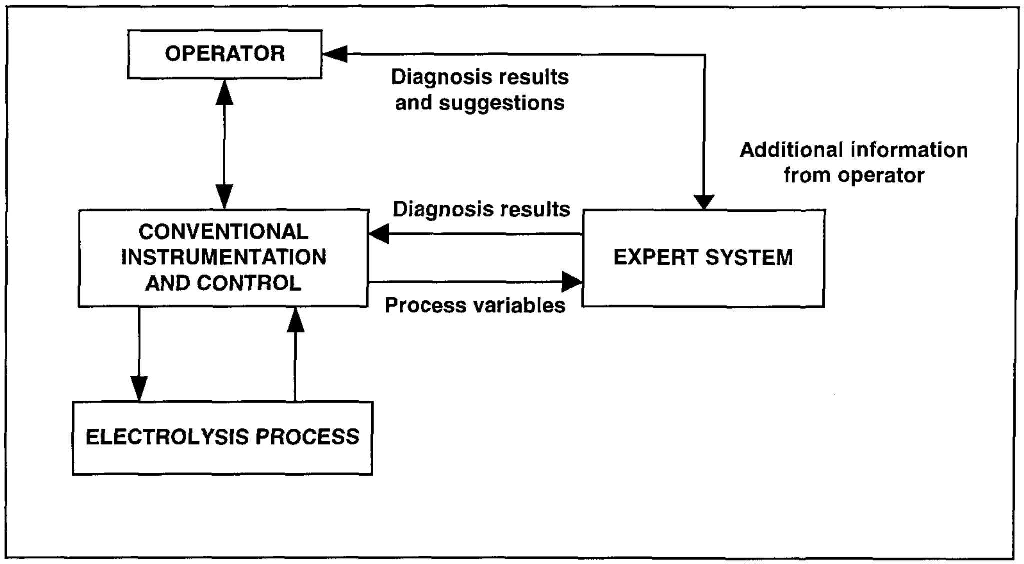

Figure 4.4 Combination of expert system and conventional control system

Therefore, to improve the electrolysis reduction cell operation, the expert system can work as a stand-alone. However, it will not only perform as an individual part but also can work together with the existing conventional control system. This combination can maintain the advantages of two systems individually without the need to change the existing control system structure. Obviously such combination can have a future in aluminum electrolysis cell process control.

In summary, when the expert system is applied to the electrolysis process, the following benefits can be obtained:

Direct measurable benefits include:

- Improved quality and better consistency
- Increased yields and reduced wastage
- Increased energy efficiency and production
- Reduced manpower through greater automation
- Reduced downtime
- Lower operating costs
- More efficient resource utilization
- Timely diagnosis of irregular cases
- Help operator to correctly solve problems
- Realize the control based on experience.

Indirect benefits include:

- Repository of knowledge
- Training spin off
- Makes expertise more widely available
- Improves competitive advantage
- Eases system understanding and maintenance
- Can lead to better working practice.

However, several limits to the use of the expert system, which restrict applications in industry are:

- Lack of understanding of domain knowledge
- Lack of skilled developers
- Difficulty to change the knowledge base
- Cost and development time

The main limitations are concerned with knowledge acquisition and expert system design. How to improve such limits is one of this work's goals.

CHAPTER 5 KNOWLEDGE ACQUISITION FOR EXPERT SYSTEM DESIGN

5.1 Introduction

In this chapter, two kinds of knowledge acquired for different purposes are introduced:

- Knowledge acquisition I – applied for knowledge engineering
- Knowledge acquisition II – applied for knowledge base casting

To develop an expert system for a specific domain application, a preliminary step is to study the fundamental knowledge of this domain. Although this type of knowledge may not be directly applied to the construction of the knowledge base, it is very important to help the knowledge engineer understand the process as it then allows the latter to organize the domain knowledge and to design an efficient expert system. The part called "Knowledge acquisition I" is collected and reorganized for this purpose. It consists of two parts:

- Relevant domain-specific knowledge
- Aluminum electrolysis production technologies

In these two parts, all the concerned domain knowledge is discussed. We also analyze the technologies of the aluminum electrolysis process, which must be considered in the expert system design, as they will directly affect the efficiency of production.

The second part of the knowledge acquisition is the knowledge that will be directly coded into the knowledge bases of AEPES (Aluminum Electrolysis Process Expert System). This knowledge concerns the real life detailed operation, as acquired from domain experts and operators.

Although these two types of knowledge are acquired for two different purposes, they can be considered as forming an entire body of knowledge.

5.2 Knowledge acquisition I – for knowledge engineering

The first thing to do in constructing an expert system is knowledge acquisition. This is often considered to be a difficult task for the following reasons:

- Domains have individual specialties. Analyzing its concepts and communicating with the experts efficiently are rarely straightforward tasks, even if the basics of the domain are well understood. For example, behind the simple jargon "anode effect," many relevant concepts are involved. If one does not have a deep understanding of the mechanism, one cannot understand how the experts solve the problems.
- The expertise and the facts presented by the experts are presented in human language. The difficulty is often in coding them in terms of a mathematical or logical function for the rule base, which is used in the programming language.

- To solve the problems belonging to a special domain, not only the relevant expertise is needed, but also the basic knowledge is required. For example, to diagnose the problems of the aluminum electrolysis process, more relevant knowledge is involved, such as information about raw materials, control system, cell design and anode quality.
- The systems applied for one type of cell are based on the knowledge with the same generality, but each individual application still has its particularities.

To solve such problems, the following steps have been applied in the knowledge acquisition process.

5.2.1 Acquaintance with aluminum electrolysis process

The domain–specific knowledge of aluminum electrolysis process could be seen as composed of several knowledge domains. Figure 5.1 shows the relationships between them.

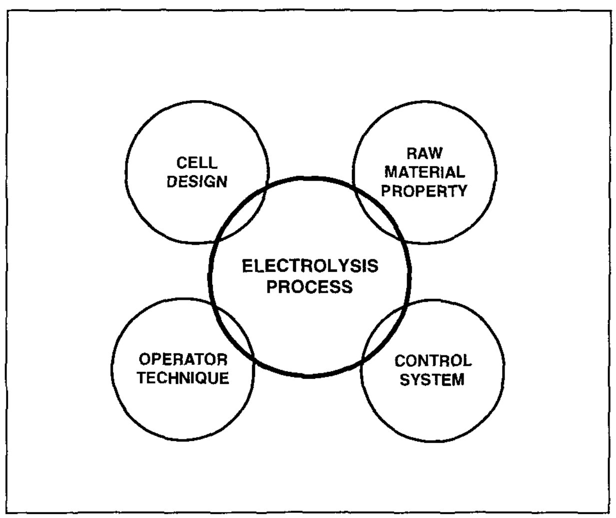

Figure 5.1 Domain-specific knowledge relationships

As described in Chapter 3, some parts of these individual knowledge domains are also the components of the expert system built in the present thesis. Therefore, in addition to general knowledge on the aluminum electrolysis process, all domain–specific knowledge should also be studied for the next step of knowledge base building.

5.2.2 Relevant domain-specific knowledge

5.2.2.1 Alumina properties

Since alumina is the fundamental raw material of aluminum electrolysis, its properties strongly affect the cell operation, final product quality and production efficiency. The important physical properties of alumina are listed in Table 5.1 [1, 2].

Table 5.1 Main physical properties of alumina

Alumina physical properties	
1	Water content
2	Specific surface area
3	Particle size distribution
4	Angle of repose
5	Flowability
6	Attrition index
7	Bulk density
8	Alpha content
9	Loss on ignition

Variations of these physical properties affect the following pot parameters:

- Current efficiency
- Energy efficiency
- Alumina dissolution
- Bath properties
- Fluoride emissions.

5.2.2.2 Process control data

The performance of aluminum electrolytic cells also depends on the cell control. A well-controlled process will lead to the higher performance of cell. The aim of cell control is to keep the process close to the targets. The controlled process parameters are:

- Cell pseudo resistance
- Pot line current
- Anode current distribution
- Alumina feeding rate
- Bath composition
- Metal and bath level.

5.2.2.3 Cell design

Different cell designs require different operating techniques that will result in different production performance. Therefore, the knowledge base must be specific to the particular cell design. For both Söderberg and prebake cells, the cell design takes into account the following aspects [3]:

- Cell current and voltage
- Busbar design

- Cell magnetism
- Cathode and anode geometry and materials design
- Automation of work practices
- Thermal balance
- Environmental protection.

5.2.2.4 Anode properties

The anode is another important raw material used in aluminum production; its quality affects the productivity and quality of the aluminum produced. Main factors monitored routinely are:

- Granulometry of the aggregate
- Amount of pitch chemical components
- Baking quality.

Many production problems are caused by poor anode properties; therefore, the knowledge of anode properties must be included in the knowledge base.

5.2.2.5 Cell operation

A skilled operator remains an important factor in cell operation. Although routine operation in modern plants is less dependent on the operators, daily maintenance, measurements and control of problem cells are still performed by the operators. The main tasks of the operator's daily maintenance work are:

- Anode changes
- Metal tapping
- Bath temperature, bath and metal level measurements
- Bath composition control
- Exception cell analysis.

The information on the cell status, gathered from operators, is required for the reasoning of the expert system. Also, the operators will carry out the suggestions given by the expert system.

5.2.2.6 Cell technologies

All new plants and most existing plant expansions are built with prebake technology. There is a trend towards larger cells and higher currents in each new generation of cells. In the early 1950s the cell current surpassed 100 kA. Thirty years later, in early 1980s, the cell current reached nearly 300 kA. In 1991, a 500 kA cell was announced. New features also include point feeders, that discharge less than 2 kg alumina per addition. All new prebake cells are also hooded for pollution control and the process control is done by computers. The operational performance of modern prebake cells has dramatically improved and is typically at 95% current efficiency and 13.5 kWh/(kg Al) of energy consumption.

5.2.2.7 New alumina feeding technology

The key to high cell performance is the alumina feeding. The most recent cell technologies use point feeding, which is done by two to four point breakers in the center aisle of the cell. Point feeding technology allows the control of the alumina concentration in the bath to a very narrow range of 1.5 – 2.5%. The lower limit is very close to the anode effect, which is avoided due to control algorithms that predict its approach. The anode effect frequency may be as low as 0.05 anode effects per cell per day. Emissions are also very small because the cells are hooded and the gases are scrubbed to 99.5% efficiency. All new plants, and most expansions of existing plants are based on this technology **[4, 5]**.

5.2.3 Basic process control technology for aluminum electrolysis

5.2.3.1 Typical process control technology

The basic function of any cell control system is to control the process variables in the short term, to make allowances for slowly changing variables and to take preventive action when abnormal cases occur in operations. The strategies of cell process control are concerned about the following **[6]**:

<u>Materials supplied</u>

- Alumina
- Aluminum fluoride
- Anodes.

Work routines

- Tapping
- Liquid level control
- Anode changing
- Anode covering.

Methods

- Consistency in working procedures
- Rectifying the deviations of the important operating variables
- Making provisions against any of the usual process disruptions or operation interference.

Control actions are based on the deviation of measured cell process variables from their target values. The following variables are controlled:

- Line amperage
- Cell pseudo resistance
- Alumina feeding
- Bath composition and temperature
- Anode cover thickness
- Bath and metal level
- Anode current distribution
- Cell pseudo resistance instability.

These variables are of great concern to two most important control strategies: control of cell pseudo resistance and control of alumina concentration.

Control of cell pseudo resistance

The cell voltage and current are the only two cell variables that are measured automatically with the sampling interval ranging from 1 – 60 seconds, depending on the technology used. Cell voltage and resistance are used to calculate the cell pseudo resistance defined as:

$$R = \frac{V - V_{ext}}{I} \tag{5-1}$$

Where: R = cell pseudo resistance
V = cell voltage
V_{ext} = extrapolated voltage
I = cell current

Cell pseudo resistance is controlled instead of the cell voltage. It turns out that the cell current variations cause cell voltage variations, but not cell pseudo resistance variations if V_{ext} is chosen correctly. Typically V_{ext} = 1.6 – 1.7 V.

Cell pseudo resistance varies with alumina concentration on a slow time scale in the order of one hour. It oscillates also when waves are present or there are other problems. The time scale of wave oscillations is from 20 – 60 s. Much more rapid oscillations with a different time scale are due to bubble release (1 s scale) or problems such as anode points, pieces of broken carbon anode in the bath, or a low anode. The cell pseudo resistance oscillations, called instabilities, are usually controlled by raising anodes, but often operator intervention is necessary in order to identify the problem.

Control of alumina concentration

The alumina concentration in the bath is one of the most important cell parameters. Unfortunately it cannot be measured. It can only be inferred from the pseudo resistance, which is deliberately varied by over and under feeding in point feeder cells. In batch fed with side broken cells, the concentration can be controlled roughly by the amount of alumina fed at each feeding. Anode effects can often be avoided by the same technique as in point fed cells.

Latest development of process control system

In recent cell process control systems, some advanced control strategies have been used which lead to substantial improvements in cell performance. One of them is the automatic control of cell heat balance. Most technologies use heat balance control algorithms based on excess AlF_3 and bath temperature measurements. The controlled variables are cell resistance and AlF_3 additions. In most recent cell designs the AlF_3 feeders are shot at intervals separated from alumina feeders. The control strategy relies on the relationship between bath temperature and AlF_3 additions [7].

5.2.3.2 Typical computer control system

At the present time, computer control systems are widely applied in the aluminum smelters. Figure 5.2 illustrates the typical structure of a computer control system.

This is a hierarchical computer control system. The supervisory computer can log and display data of the whole plant. At the intermediate level, the control computer monitors line current, cell voltage and pseudo resistance and all other measured variables. It also provides access to data and process analysis software. Some technologies have individual cell controllers, which control cell resistance and alumina feeding and the cell state control box. The cell controllers are mounted on the wall near each cell and are connected to the central control system. The potroom operator can select between automatic and manual control. The cell state communicator activates special control routines such as metal tapping and anode changing.

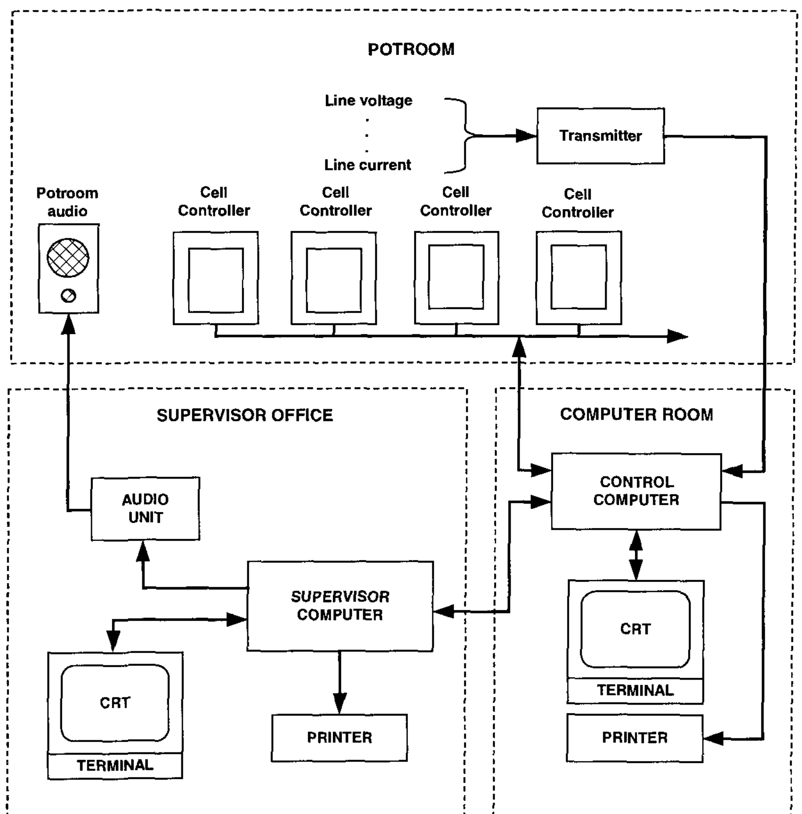

Figure 5.2 Typical structure of a smelter computer control system [6]

5.3 Knowledge acquisition II – knowledge casting to rules

Most sources of knowledge used for the present thesis are published papers. Several open sources are studied. The most important ones are:

- "Diagnosis and Correction of Irregularly Operating Cells" by Tabereaux [9]. The basic knowledge of an engineer level expert system is taken from this source.
- Horvath's special report on the detection of typical disturbances in the operation of all types of cells. This contribution is of benefit to our operator level expert system [10].
- Master's thesis of DesBiens, whose knowledge base was developed for alumina feeding of the Alcan high amperage experimental cell (A-310) [11].
- Two other expert systems were analyzed as reference material. Both were built for diagnosis of anode problems in a vertical stud Söderberg cell [12].
- Since it is desired that our expert system should be used in conjunction with the existing control system, the general knowledge of existing process control systems of aluminum electrolysis was also studied.

In the following sections, we will analyze each of the aforementioned knowledge sources.

5.3.1 A special course materials

The general knowledge applied in our system is collected from a special course of Alton T. Tabereaux: "Diagnosis and Correction of Irregularly Operating Cells, Aluminum Electrolysis - Theory and Practice of Primary Aluminum Production". We began analysis on the version 1992 of this course. The fundamentals of ENGES knowledge base were built with this knowledge. This course material describes diagnosis methods for irregularly operating cells. We updated this material with two latest versions of the course **[13, 14, 15]**.

5.3.1.1 Basic structure of the course

Most of the knowledge found is of a general nature and is not related to a specific cell or class of cells. It is divided into five classes and related sub classes. Our knowledge base also follows the same arrangement and uses similar classes and sub classes:

<u>Unstable cell voltage</u>

- Low anode
- Excessive metal tapped from cell
- Carbon lump in bath - under anode
- Broken anode carbon
- Broken stem or anode burn off
- Anode heeling - on lumps
- Metal "roll" or inversion
- Spiked or grounded anode.

<u>Excessive number of anode effects</u>

- Empty bins or obstruction of bins and feeders
- Liquid bath level too low
- Bath temperature too low
- Change in alumina properties.

Muck (sludge) accumulation in the cathode

- Soft alumina slurry / or muck
- Hard thick muck - very hot "sick" cell.

Anode carbon quality

- Anode quality
- Cell operations.

Higher iron impurity

- Anode stem erosion
- Cathode collector bar and cast iron.

5.3.1.2 Basic characteristics of the course

In order to code Tabereaux's knowledge into rules and to build a knowledge base with an efficient reasoning ability, it is necessary to extract the basic characteristics of his knowledge. Several characteristics of this knowledge can be identified:

Instability of cell voltage

Computer technology has greatly improved the control and performance of cells. In the computer control approach, the variation between the cell voltage measurement and designated set point value determines the relative instability of the cell. Therefore, Tabereaux uses the instability of cell voltage to provide early warning information to potroom operators and to diagnose the most common operational problems. He also uses the standard deviation of the cell voltage as a voltage stability index over different time periods and provides short or long-term instability information. No doubt, this unified instability of cell voltage will help us build some rules, which will lead to faster reasoning and to a clearer questionnaire of the user interface.

Visual descriptions of process problems

The aluminum electrolysis process is a complex electrochemical process and there are many problems occurring in the small space around the anode-cathode. In many cases, there is no obvious visual phenomenon to be seen from the outside. To help the operators identify the different cases, Tabereaux provides the graphics of sick cells.

Symptom-Cause-Action presentation

A rule-based system is our choice for the expert system design. In Tabereaux's work **[9]**, the symptom-cause-action presentation is close to the "IF-THEN" structure of a rule. The "symptom" presents the various phenomena occurring in a problem, the "cause" specifies diagnosis results and the "action" tells the operator how to solve the problem. Such a straightforward and natural presentation helps the operator understand, and is also of benefit to our rule base construction.

5.3.1.3 An example of diagnosis and correction

Table 5.2 is a typical example with the structure "Symptom-Cause-Action" and its graph:

Table 5.2 An example of diagnosis and correction

Problem	Low anode	
Symptoms	1	Computer instability factor is higher than normal
	2	Individual anode currents are high and more unstable
	3	Anode spike, red stem or anode burn off may occur
Causes	Anode is too low	
	1	Anode set too low
	2	Increase in cell resistance
	3	Increase in metal turbulence
Actions	1	Measure anode current distribution to determine which is the faulty anode
	2	Raise anode until a normal current and stable cell voltage result
	3	Recheck current about two hours later. Including all anodes in cell
	4	Ensure correct anode setting on new anodes
Graphics	1	Computer instability graphic chart (Figure 5.3)
	2	Diagram of low anode case (Figure 5.4)

Two graphics listed in Table 5.2 are shown next as Figure 5.3 and Figure 5.4:

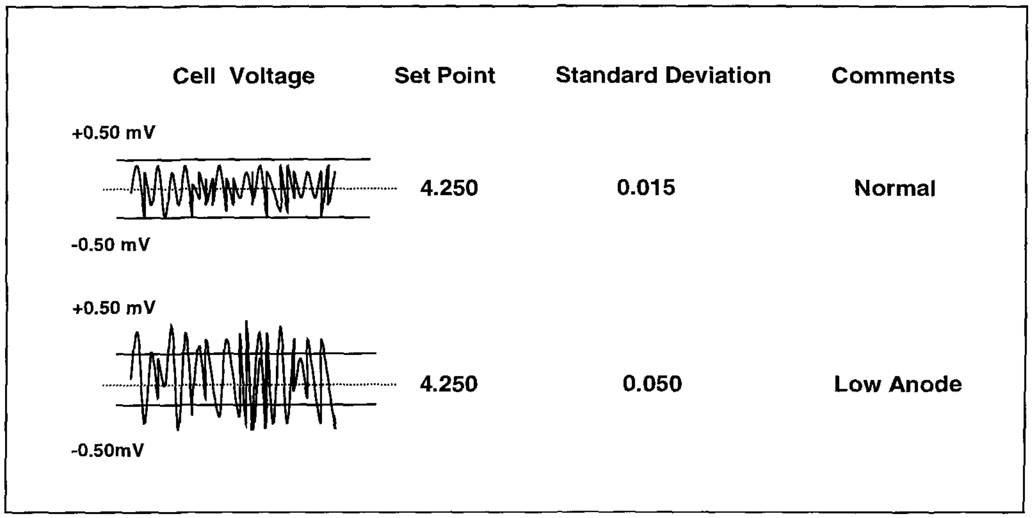

Figure 5.3 Computer instability graph charts

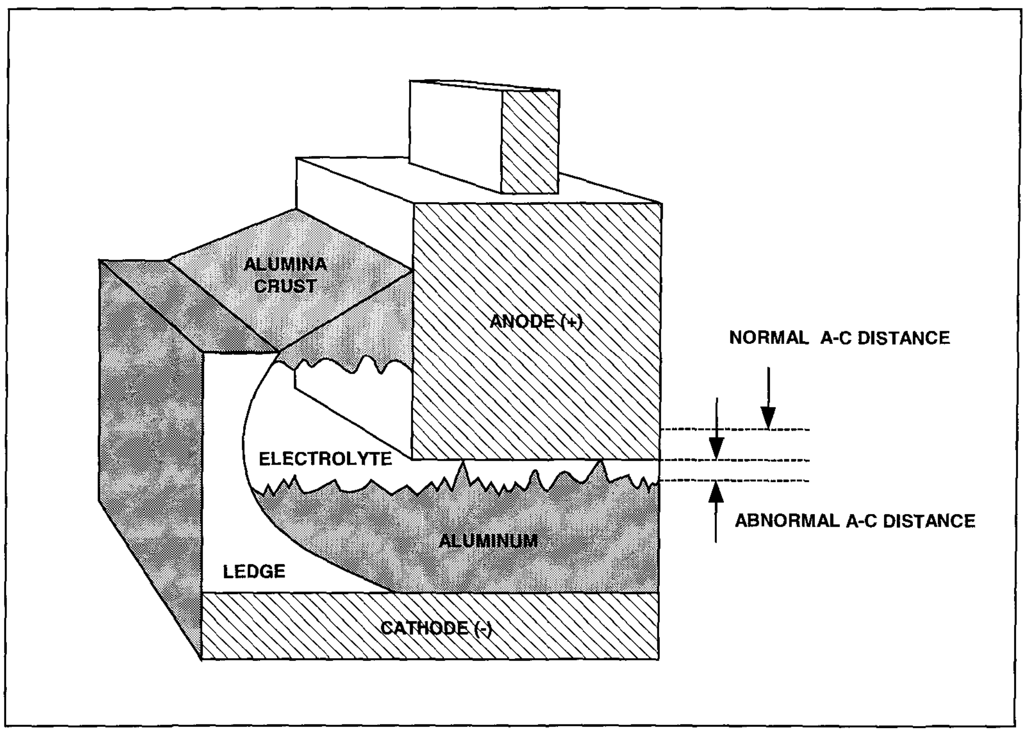

Figure 5.4 Low anode case

5.3.1.4 Conclusion

Depending on the analysis of the knowledge from Tabereaux, we found that the characteristics, the structure of the knowledge and its diagnosis function are close to that needed for our expert system.

In six sub-systems of the knowledge base of ENGES (See **Section 6.3.1 "Structure of ENGES"**), the first five sub-systems are mainly adopted from the Tabereaux's course. The remainder of the knowledge is acquired from other sources.

The five sub-systems of ENGES, based on Tabereaux are:

- ENGES_1.knw: 20 rules
- ENGES_2.knw: 35 rules
- ENGES_3.knw: 14 rules
- ENGES_4.knw: 15 rules
- ENGES_5.knw: 14 rules

5.3.2 An internal report to UQAC

Horvath summed up his expertise in a special report to UQAC **[10]** as a contribution to our knowledge base. Whereas Tabereaux's course is of a general nature and not related to a specific cell, Horvath's report is more concerned with specific cell problems. The major topics of this report are the following:

- Narrow range of variations
- Consistent operation
- Operation of a pot-line as a single unit

- Adjustment of common parameters to maintain the proper heat balance.

Horvath also provided a reference/test data bank for four types of cells.

5.3.2.1 The basic structure of the report

The knowledge provided in this report is based on the following typical operational states and disturbances in the cell operation:

- Cooling trend in cell operation
- Warming trend in cell operation
- Typical disturbances in cell operation.

In view of solving the routine operation problems, the major part of this knowledge is the analysis of typical disturbances in cell operation. The typical disturbances are grouped into two major aspects:

1. Operational concerns
    - High energy consumption
    - Low production
    - High raw materials consumption
    - High labor load
    - Low efficiency operation
    - High pollution.
2. Cases identified

The detailed contents of the cases identified are described in Table 5.3, Table 5.4, Table 5.5, and Table 5.6:

Table 5.3 Case 1: Detection of disturbances based on data supplied from pot controllers

Case 1	Identified from pot controllers
	• Unstable cell voltage, the cell voltage fluctuation is lower than normal • Unstable cell voltage, the cell voltage fluctuation is higher than normal • Anode effect frequency is lower than normal • Anode effect frequency is higher than normal • Average anode effect voltage is lower than normal • Average anode effect voltage is higher than normal • Number of daily anode movements is less than normal • Number of daily anode movements is more than normal

Table 5.4 Case 2: Detection of disturbances based on data supplied from traditional plant measurements

Case 2	Identified from traditional plant measurements
	• Electrolyte level is higher than normal • Electrolyte level is lower than normal • Metal level is higher than normal • Metal level is higher than normal

Table 5.5 Case 3: Detection of disturbances based on data supplied from laboratory analysis

Case 3	Identified from laboratory analysis
	• Iron content increases in the metal • Silicon content increases in the metal

Table 5.6 Case 4: Detection of disturbances based on data supplied from non-traditional plant measurements and observations

Case 4	Identified from non-traditional plant measurements and observations
	o Anode problems o Sludge problems o Ledge profile problems – poor ledge o Ledge profile problems – extended ledge o Red anode stem (Prebake cell) o Intense anode top evaporation (Söderberg cell) o Anode covering by alumina (Prebake cell) o Sticking between anode casing and carbon anode (Söderberg cell) o Anode gas burner problems o Gas hole and feeding hole problems o Gases escaping through crust

5.3.2.2 Testing databank for selected cell types

In order to test our knowledge base, a databank was compiled for four types of cells:

- Low amperage, Söderberg, vertical stud (VSS)
- Low amperage, open prebake pot (OPB)
- Medium amperage, hooded prebake pot (MPB)
- High amperage, hooded prebake pot (HPB)

This databank contains two types of data in sub databanks. The first is the general pot line data, as well as a brief description of the cell type. The second is the process data from specially selected cells and cases. All the data in the databank came from four possible sources:

- Cell controller
- Regular measurements taken by potroom operators
- Observations (and no regular measurements) taken by potroom operators
- Data from laboratory analysis.

5.3.2.3 Designated parameters for the examined cell types

For on-line diagnosis and consultation, the following designated parameters for the steady state of four typical cell constructions and cell operations are also proposed:

- Line current
- Electrolyte composition
- Electrolyte temperature
- ACD
- Anode voltage drop
- Cathode voltage drop
- Bus bar voltage drop
- Current efficiency
- Daily production
- Alumina demand per day
- Scheduled crust-breaking interval
- Number of anode effects/day
- Alumina feed at anode effect
- Target value of 48h tapping interval.

5.3.2.4 Basic characteristics of the report

Compared to other sources, some different characteristics of this report are:

- The knowledge generalized from Horvath's 30 years experience in the domain of aluminum electrolysis process helps us create very concrete suggestions for the operators.

- The presentation of the fault diagnosis process in Horvath's report also uses symptom-cause-action structure for special cells that is similar to our rule structure.
- Data collected directly from the operating plants provides the possibility to verify the prototype of our knowledge base during the development.
- In the preliminary stage, for the different expert system shells (Comdale and Gensym's G2), the databank was developed using MS Excel. One sheet of an Excel file contained the general pot line data. Several separated sheets contain the data for specific cases. The Excel file was used as a type of databank, which allows the expert system to communicate through the network by DDE or ActiveX mode.
- Detailed descriptions of symptoms of cooling and warming trends in operation provide relevant knowledge for long term or short term cell status prediction.

5.3.2.5 An example of diagnosis and correction

A typical example from Horvath's report **[10]** identifies the causes of unstable cell voltage, which is a common problem in production. This example presented the formalized structure and suggested execution instructions. These are shown in Table 5.7,Table 5.8,Table 5.9, Table 5.10,and Table 5.11.

Table 5.7 Causes of unstable cell voltage and cell voltage fluctuation is higher than normal

Cause Number	Causes Identified
Case 1	Anode cathode distance is too low
Case 2	There is short circuit between the anode and cathode
Case 3	Electrolyte and metal are mixed

Table 5.8 Suggested action for cause No.1

Cause No.1: Anode cathode distance is too low	
Action	Increase the cell voltage by 100 mV, then check cell operation after 1 hour

Table 5.9 Detail causes of cause No.2

Cause No.2: There is short circuit between the anode and cathode	
	Detail causes
Case 2.1	Metal waving, too much metal was tapped from the cell
Case 2.2	1. Carbon particles or carbon pieces in electrolyte 2. Poor anode quality or failure operation
Case 2.3	1. Anode spike 2. Poor anode quality or failure operation
Case 2.4	1. Anode is on the frozen ledge 2. Operation failure and extended ledge

Table 5.10 Suggested actions for detail causes of case No.2

Actions for detail causes of case No.2	
For Cause 2.1	1. Check the metal height • Add hot metal into the cell from another cell, if needed • Skip the next tapping 2. Check the cell voltage stability after 2-4 hours
For Cause 2.2	1. Check the anode quality (laboratory test). Use a quality anode that meets the specifications 2. Remove the carbon particles or carbon pieces from the electrolyte 3. Check the cell voltage stability after 2 hours

For Cause 2.3	1. Check the anode bottom to find the location of anode problem. If spike present, lift the anode and remove the spike. 2. Check the cell voltage stability after 2 hours
For Cause 2.4	1. Check the anode bottom to find the location of anode problem. If spike present, lift the anode and remove the spike. 2. Check the cell voltage stability after 2 hours

Table 5.11 Actions for case No.3

Case No.3: Electrolyte and metal are mixed	
Action 1	Lift the cell voltage by 200 mV
Action 2	Add hot electrolyte into the cell
Action 3	Increase the alumina layer thickness on the crust
Action 4	Check the cell voltage stability after 3-4 hours

5.3.2.6 Conclusion

The knowledge acquired from Horvath constitutes the main part of the rule base of OPEES. Some of this knowledge was also used in ENGES.

The knowledge used in ENGES is mainly in the domain of "Trends Prediction." There are 20 rules coded into this part of the rule base. Some detailed suggestions about pot operation are also built into the relevant rules of ENGES.

In OPEES-1 and OPEES-2 (See **Section 6.3.2 "Structure of OPEES"**), there are four sub-systems in each. In each sub-system of OPEES-1, 85% of the total of 298 rules are based on the knowledge of Horvath. But for the sub-system of OPEES-2, there are also 85% of total 302 rules based on Horvath's knowledge. Depending on these rule bases, OPEES has some ability to diagnose the on-line information and to do the reasoning to identify the problems.

5.3.3 A master's thesis about an alumina feeding expert system

This master's thesis of DesBiens **[11]** provided a design of an off-line expert system for an experimental electrolytic cell A-310 of the Arvida Research and Development Center of ALCAN.

5.3.3.1 Basic structure of the knowledge base

This expert system was developed using the Comdale/X expert system shell. To construct the knowledge base for this cell and its special feeding system, DesBiens collected the knowledge from the following five sources:

- Devices and mechanical parts of alumina feeding system
- Computer system
- Anode effect properties
- Cell pseudo-resistance tracking
- Sludge formation in the cell
- Interviews with cell experts and operators.

Based on the properties of different objects of the knowledge collected, sixteen classes are used to identify such different objects. Among them, thirteen classes are concerned with the following six aspects of the feeding system:

- Air-slide conveyor system
- Feeder
- Alumina hopper
- Feeder test
- Alumina properties
- Breaker.

5.3.3.2 Basic characteristics of this work

Interviews with experts

In order to become acquainted with the aluminum electrolysis process and to collect relevant knowledge, one important method adopted by DesBiens is interviewing the domain experts. The following topics were discussed:

- Alumina feeding schedule and transport system
- Cell voltage and resistance
- Alarm and computer system
- Anode effect
- Sludge problem
- Relationship between alumina feeding and bath level.

Strong points of this work

- It can be used to diagnose the problem or help to make a decision for a real cell (Cell A-310)
- The question uses natural language, which facilitates the interaction with the operators
- Interfaces consist of both text and graphics for easy understanding.

Decision tree

DesBiens created a decision tree to represent the knowledge acquired from experts and used it for rule induction and for solution searching. A decision tree is a graphical representation of a procedure for classifying or evaluating the elements of its domain. For large and complex decisions, the decision tree can organize the elements efficiently while considering all possible options.

Although the decision tree may be valuable for the diagnosis of some problems, for most rule-based expert systems, it is not the typical reasoning approach. Forward chaining and backward chaining are still two basic technologies used in such cases.

5.3.3.3 Conclusion

Although the performance and the application range of this expert system were limited by its task and the software applied, we still can find some benefit from its knowledge base. First of all, the detailed description of the conversations, which contains the experience from domain experts and operators, helps us to become acquainted with this domain knowledge and to learn how to acquire the knowledge from interviews with the domain experts.

Secondly, some knowledge about the alumina feeding system can be used in the knowledge base of ENGES. This knowledge is built into five rules of the second sub-system of ENGES. But it is clear that we cannot directly apply the knowledge due to its specific source, which is the A-310 cell. We selected only the knowledge related to the alumina feeding system and the number of anode effects.

5.3.4 Two relevant expert systems

Two expert systems in the domain of the aluminum electrolysis process, EPURNXP and VS-Anode Expert System, were made available to us **[12]**.

These two systems were developed using the NEXPERT expert system shell. We tested these two applications, but unfortunately, due to some necessary files missing we could not run the EPURNXP, and only the VS-ANODE Expert System was examined **[16, 17]**.

This expert system was designed for the Söderberg anode problem diagnosis. From the rule base analysis and the testing results, we found that there are four main problems addressed in this knowledge base:

- Anode dusting problem
- Manufactured materials
- Paste leaks
- Raw materials.

The reasoning process is based on the specific values of anode parameters. The VS-ANODE Expert System was built only for Söderberg anodes in an ALCAN smelter. The parameter values used are also specific. Only with these can the diagnosis be found. For example, while the anode dust problem is checked, the question asked is: "What is the value of the pitch mesophase content? The normal value ranges from 0.0 to 1.0%. Please type the value." For such kind of problems, if the required parameters are not available, the diagnosis process has to be discontinued. Therefore, it is impossible to apply such knowledge directly to our expert system.

The VS-ANODE Expert System does use some text questions instead of numerical parameters. For example, a text question shows that: “Is the pitch_content_high status?” No numerical parameters are needed for this kind of question. Unfortunately, such text questions cannot generate the final reasoning result as only preliminary suspected results are provided. If we want to obtain a final reasoning result, additional numerical parameters are required. For example, the conclusion will be: ”Problems with the paste aggregate fraction are suspected. Please enter the detail fractions data.” In a total of 147 rules of the VS-ANODE rule base, only 21 rules use text questions. In order to benefit from the knowledge of the VS-ANODE Expert System, we selected 19 of these rules and transposed them to ENGES for reference only and not as a standard function selection.

More information about the NEXPERT and the detailed analysis about the VS-ANODE Expert System can be found in **APPENDIX 2 “VS-ANODE Expert System Study”**.

5.4 Acquired knowledge management

After studying various sources and interviews with the domain persons, the collected information still needs to be analyzed. The objectives of this effort are to determine what was learned and what additional information should be acquired.

Then the individual knowledge has to be integrated into the overall knowledge base.

5.4.1 Acquired knowledge analysis

Knowledge collected from domain persons will provide confidence in the future behavior of the system, but there can never be total certainty about the correctness of any experience. The quality of knowledge acquired will often be the determining factor in its success.

According to the requirement of the particular purpose of the expert system, the acquired knowledge must be analyzed and organized, and then integrated if it comes from different sources. For instance, Tabereaux's knowledge is used for the general analysis of the electrolysis process, but Horvath's findings make more sense for detailed operating practices. Both of them give different views for the fault diagnosis process. Therefore, the final knowledge bases were connected with each other. But for the different purposes, individual knowledge bases still keep their characteristics.

5.4.2 Acquired knowledge organization

It has been said that we analyzed the knowledge acquired and organized it into two expert systems: *engineer level* and *operator level* expert systems. First, we built the engineer level expert system, ENGES, which mainly based on the knowledge from Tabereaux's course. Secondly, when Horvath's report was made available, we developed the operator level expert system, OPEES. The knowledge base and the database are coded for the different types of the cell and on-line consultation. The final knowledge bases compensate for the personal limitations of the individual experts by drawing from each other's sources.

5.4.3 Knowledge maintenance

From the start, knowledge maintenance is a considerable problem. The knowledge evolves; therefore, the modification of the knowledge base becomes an unavoidable task, which has to be concerned with the following aspects:

- Over time, the behavior of the cells changes gradually. To obtain correct diagnosis results, the related database must be modified. For example, the previous diagnosis results show that there is a small hole in the cathode of the No. 45 cell. If such problem remains there, then this information must be noted in the database to help further diagnosis.
- An expert system often faces updating of hardware and software.
- User evaluation and acceptance of the expert system will be heavily influenced by the quality of the diagnosis process. During the normal operating process, the user will often offer some new elements to the knowledge.
- In order to simplify the routine maintenance work, the interface must be designed so that the data can be easily modified. For example, the alarm limitations of process data are different for different cell ages. These parameters should be easy to change without modifying the rule base.

This points to the fact that if an expert system is to be useful in an industrial application, the system must be designed for easy maintenance.

CHAPTER 6 AEPES – BASIC DESIGN

6.1 Objective of basic design

The objective of this work is to develop an expert system for diagnosis of irregularly operating aluminum electrolysis cells. As mentioned previously, there are several types of cells in primary aluminum production. Each type requires different operating practices. Normally, an expert system would be built for each cell technology. Until now, no generalized expert system for all types of cells has been built. Our challenge is to build one expert system for several technologies and at the same time to allow for specific features of each cell type. The expert system must also be easy to maintain. It is hoped that this expert system will benefit industrial users, particularly those for whom human experts are not available.

6.2 Basics of Comdale/X

The expert system shell applied for the basic design of AEPES is Comdale/X, which is a consultative expert system. It is principally used for diagnostics, training, intelligent on-line documentation and operator support. Comdale/X provides a user-friendly environment for the development and delivery of expert system applications used in stand-alone or embedded applications. Embedding is the key element of Comdale/X as it allows use of existing control software and the host system. Comdale/X consists of two developing components, an expert system editor and a form editor, applied for system development and interface design.

The new generation shell of Comdale, SmartWorX, was also one of the strong candidates as new developing shell. We have tested SmartWorX for verification and comparison. Some preparations for software transformation have been done. The detail information and application example of Comdale/X and SmartWorX can be found in **APPENDIX 1.1 "Comdale/X"** and **APPENDIX 1.3 "Application examples."**

Finally, although SmartWorX was not applied to develop our expert system, we learned many things from this test procedure that will benefit our system in many aspects, such as: system design, intelligent alarm system, interface and real-time function design.

6.3 General structure of AEPES

The management of generalized knowledge and particular knowledge in a single integrated expert system is difficult. If the knowledge base is built with the general knowledge, it will be used for solving the general problems only. For particular cells, more particular domain knowledge for individual cells has to be included. We built a two-level expert system named AEPES (Aluminum Electrolysis Process Expert System): the engineer-level sub-system is named ENGES (Engineer Expert System) and the operator-level sub-system is named OPEES (Operator Expert System). The generalized knowledge about the aluminum electrolysis process is entered into the former. The latter does not only contain the generalized knowledge, but also complements the cell-specific knowledge. The knowledge is organized into different modules, which are the building blocks of the knowledge base. The advantage of this modularized approach is its flexibility as it will be easier to develop a knowledge base for a particular cell, while adding the special knowledge modules to the general knowledge.

The relationship of these two expert systems can be compared to the tasks of the engineer and the operator:

- The engineer expert system is concerned with general analysis.
- The operator expert system makes quick decisions and provides detailed suggestion on operational practices, similarly to a skilled operator.

AEPES can operate in two modes: consultation (off-line) and real-time (on-line). The real-time operation is used in conjunction with the cell simulator, which generates the data. The operational mode is selectable by the user. Components of AEPES can work co-operatively or separately. It should be noted that ENGES could work in off-line mode only whereas OPEES can work in both modes. The general structure of AEPES is given in Figure 6.1.

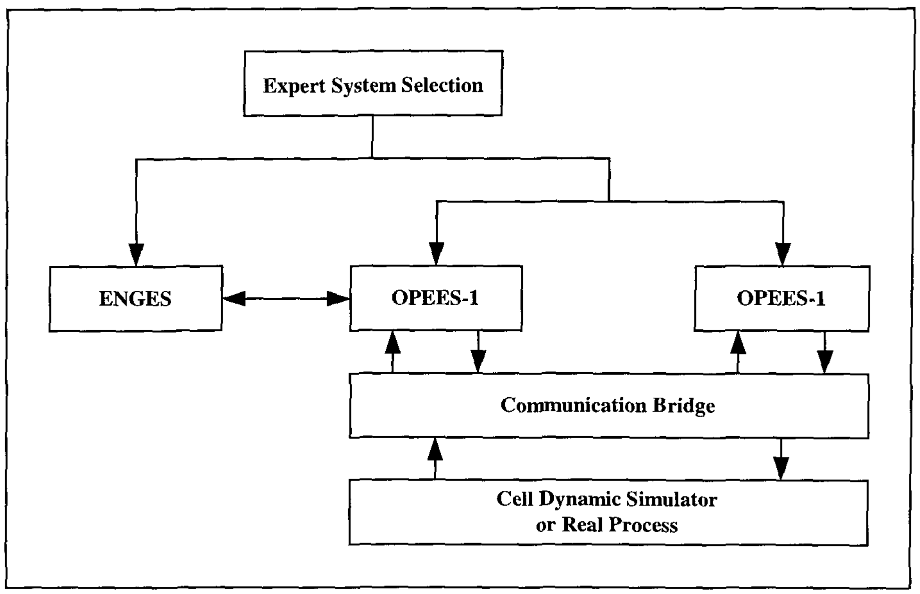

Figure 6.1 Structure of AEPES

ENGES is normally used in analysis or as a training tool for either engineers or operators.

ENGES can co-operate with OPEES. When they co-operate, the results and predictions of ENGES are transferred to OPEES or are saved in a file. The results transferred to OPEES increase its speed and accuracy. Subsequently, the results of OPEES can be fed back to ENGES. The system and process data of ENGES are modified by this feedback information. When OPEES works in the on-line mode, two types of OPEES can be selected for two different cases, as will be specified later in **Section 6.3.2 "Structure of OPEES."**

The user interacts with the AEPES via the user interface. Several forms have been designed for questions and answers. Although the user has access to the components of the knowledge base (classes, objects and rules) and to the inference engine (to select different reasoning strategies), these actions are better reserved for the knowledge engineer.

ENGES is built around the general knowledge gained from the work of Tabereaux [1, 2, 3, 4] and OPEES applies information taken mainly from particular knowledge and detailed process data related to specific cells, acquired mainly from the work of Horvath [5].

6.3.1 Structure of ENGES

The structure of ENGES is shown in Figure 6.2. When AEPES is started, the user can select "ENGES" from the "Expert System Selection" form for running.

Based on the nature and structure of the general knowledge collected, the knowledge built into ENGES is divided into six classes. These are:

- Unstable cell voltage
- Excessive number of anode effects
- Muck (sludge) accumulation in the cathode
- Anode carbon quality
- Higher iron impurity
- Trends prediction.

The user can select the problems from those six classes to start the reasoning. As ENGES works in off-line mode, all the information required should be entered via the user interface. ENGES will ask relevant questions about the selected problems to acquire the necessary information for the reasoning. The user can answer the questions with "YES", "NO" or "UNKNOWN." If the answer cannot be confirmed, the user can use the degree of certainty to tell ENGES how sure the answer is. During inference, the inference engine will organize all the entered information and then the inference strategy is applied to find out how and when the reasoning session is to end. The inference strategy is used by both forward and backward chaining activities. When the reasoning process ends, the diagnosis results and suggestions are displayed. The suggestions list the recommended actions to help the user solve the problem at hand.

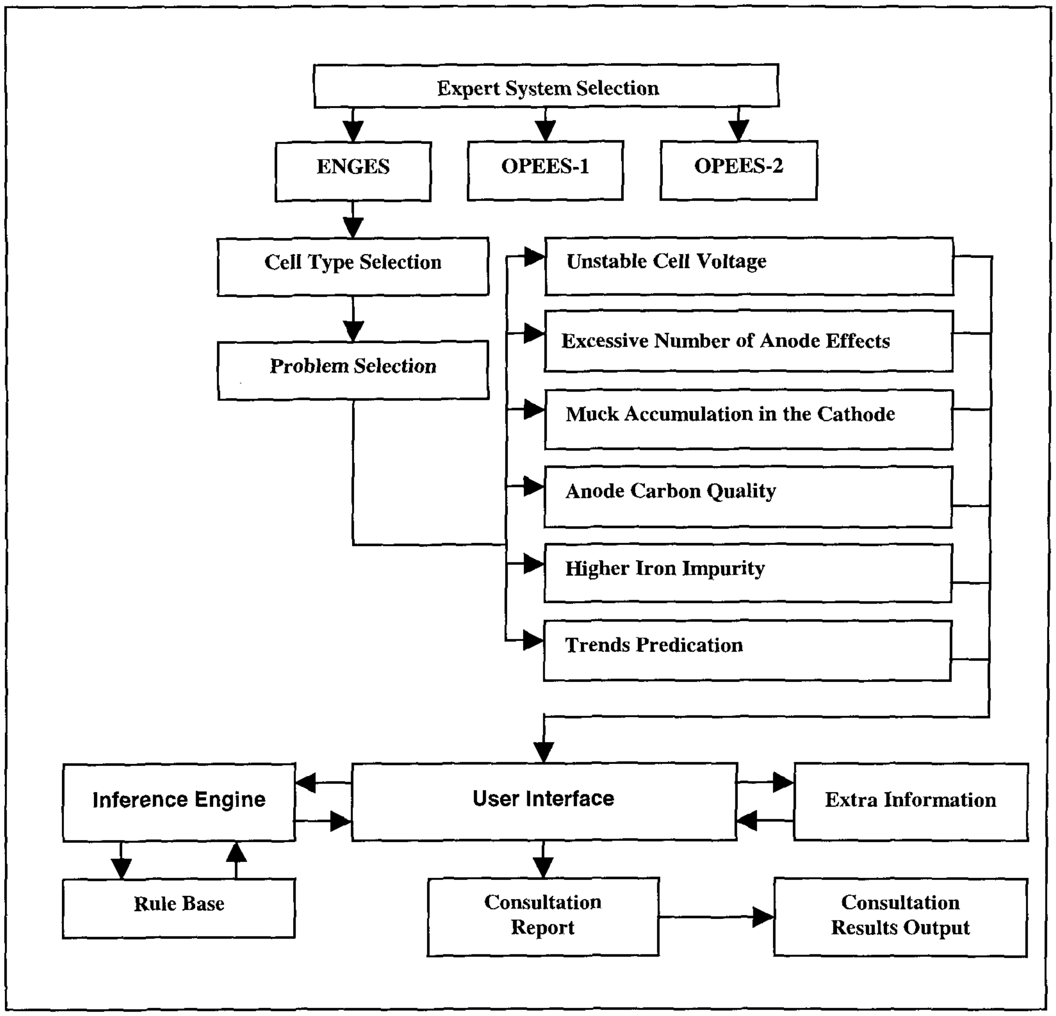

Figure 6.2 Structure of ENGES

6.3.2 Structure of OPEES

There are many types of electrolysis cells in currently operating aluminum smelters. Their process control systems are also different. In order to work with the different cases, we designed two sub-systems named: OPEES-1 and OPEES-2 for typical types of cell control systems. OPEES-1 works on an advanced case where most of the process data are available in a database. OPEES-2 is employed when only limited process data are available. This case is close to the real situation prevailing in aluminum smelters.

The structure of OPEES is given in Figure 6.3. To start OPEES-1 or OPEES-2, the selection has to be done in the "Expert System Selection" form. For a particular cell, the cell type, cell line and cell number must be selected.

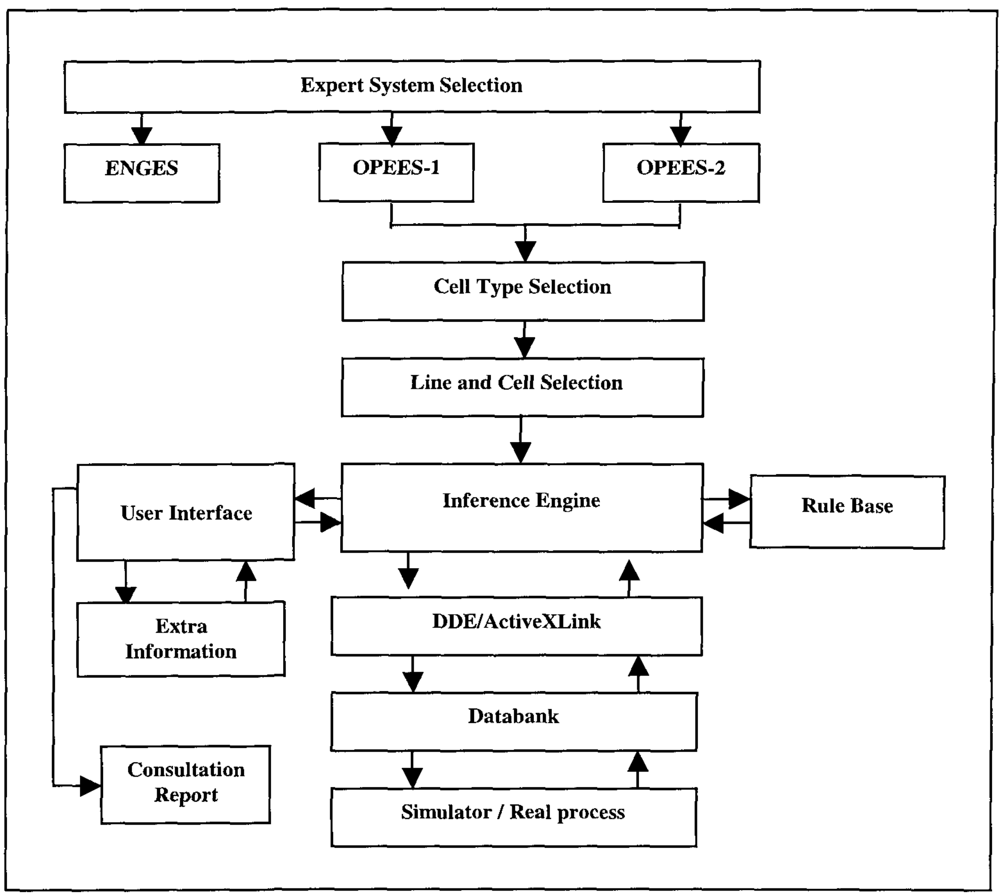

Figure 6.3 Structure of OPEES

In the on-line mode, OPEES monitors all incoming process data, and then releases the current alarm status, if any exists. The user can select any "ON" alarm status to force OPEES to start the relevant diagnosis. Depending upon the nature of the alarm states, OPEES may ask the user to provide further information that is not available in the on-line databank. When the reasoning process ends, OPEES provides the diagnosis results and suggestions. The user can study those suggestions in order to check or adjust the faulty cell.

The applied data communication depends on the shell selected. Comdale/X supports only Dynamic Data Exchange (DDE), which is a standard inter-process communication protocol for Windows and Windows NT. Comdale/X supports two types of DDE conversation: MS Excel and ORACLE. In our applications, the Excel-type was used. Further, we used Net DDE to enable the data transfer from the simulator or from the real process to the data files. Net DDE refreshes the data files by selected cycles and OPEES monitors them. There are three types of data in the data files: static data (e. g. set point), dynamic data (e. g. process data) and inference results (e. g. decision).

OPEES can be applied for four typical types of cells:

- Low amperage Söderberg, vertical stud (~75 kA)
- Low amperage, open prebake pot (~75 kA)
- Medium amperage, hooded prebake pot (~185 kA)
- High amperage, hooded prebake pot (~300 kA)

To test and to verify OPEES in the on-line mode, a virtual cell, developed at UQAC was used **[6, 7, 8]**. The benefits of a virtual cell are:

- It can simulate different operational states of selected aluminum electrolysis cells
- The applied controller emulates a general-purpose, multi-level, distributed control system
- The interface acts as an intermediary between the operator and OPEES.

6.3.2.1 Structure of OPEES-1

OPEES will work in the on-line mode for particular types of cells. Based on analysis of the current cell control systems, two types of OPEES were designed. The major difference between them is the number of available process data. OPEES-1 can directly access the real-time database through the communication bridge. In our laboratory, the virtual cell supplied most of the process data. Normally, the main process data of the reduction process are as in Table 6.1:

Table 6.1 Main process data

No.	Major process data
1	Cell voltage
2	Cell voltage deviation
3	Cell voltage fluctuation
4	Number of anode effects for last day
5	Average voltage of last anode effect
6	Maximum voltage of last anode effect
7	Duration of last anode effect
8	Number of anode adjustments per day
9	Cathode voltage drop
10	Weight of tapped metal
11	Metal height before tapping
12	Bath height
13	Bath temperature

14	Anode current distribution
15	Cathode current distribution
16	Ledge profile
17	Red anode stems
18	Anode gas burner operation
19	Color of the flame
20	Fe content
21	Si content
22	Bath ratio

We chose twelve of the most important process data from them as the on-line monitored process data of OPEES-1. Table 6.2 is the list of the monitored process data chosen.

Table 6.2 Monitored process data

No.	Monitored process data
1	Cell voltage fluctuation
2	Number of daily anode adjustments
3	Average voltage of last anode effect
4	Number of anode effects
5	Iron impurity content in the metal
6	Silicon impurity content in the metal
7	Bath ratio
8	Ledge profile
9	Bath level
10	Bath temperature
11	Metal level
12	Weight of tapped metal

Using the alarm form, which was designed for monitoring these parameters, the operator not only can obtain the status of out-of-bounds process data, but also can choose one of them to make the relevant diagnosis. According to the status of selected data, the inference engine will start the first stage reasoning. If more information is needed, a check terms list form will appear and ask the operator for the missing data, some of which may still have to be measured.

Finally, the diagnosis is given and the corrective action is suggested. The structure of OPEES-1 is shown in Figure 6.4.

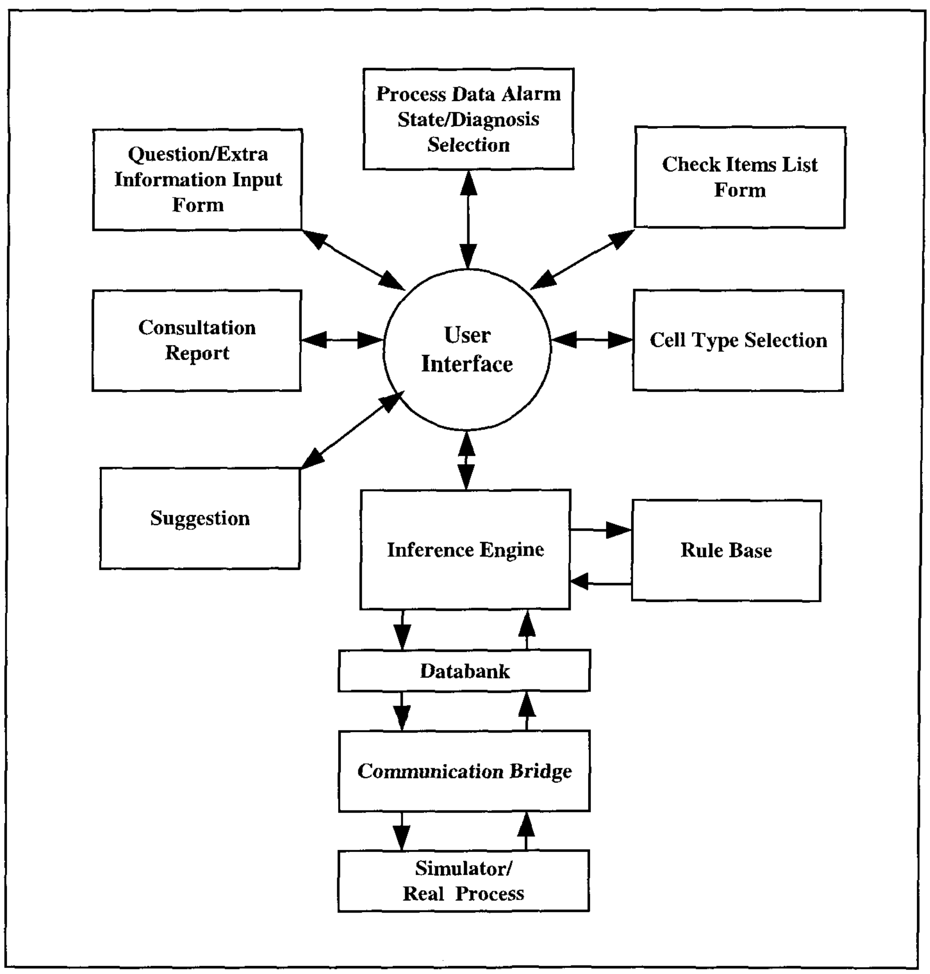

Figure 6.4 Structure of OPEES-1

6.3.2.2 Structure of OPEES-2

It has been previously stated that only a few process data can be measured continuously. In order to meet this requirement, we designed OPEES-2. The following parameters were chosen for OPEES-2:

- Cell voltage fluctuation
- Number of daily anode adjustments
- Average voltage of last anode effect
- Number of anode effect.

It is impossible to do reasoning with incomplete information. Additional information should be obtained through the user interface. A special double reasoning strategy is used for OPEES-2. The first reasoning is based on the automatically monitored process data. Then, an inter-diagnosis report will tell the operator what additional information is needed and where to obtain them. OPEES-2 releases a checklist and asks for all required information at the same time.

6.4 Knowledge base building

6.4.1 Knowledge base of ENGES

Characteristics of the knowledge base of ENGES

As mentioned above, the primary purpose of ENGES is an application for the general aluminum electrolysis process. The major characteristics of the knowledge base of this expert system are:

- Applied for general purpose diagnosis
- Generalized knowledge suitable for most types of cell
- Easy to compile
- Can be used as a training tool.

Based on such characteristics, we acquired the corresponding knowledge in two ways:

- General knowledge from books, references, and industrial visits
- More detailed knowledge from domain experts.

Main components of the knowledge base of ENGES

According to the application purpose of ENGES, the structure of knowledge base was mainly built on the analysis of the Tabereaux's course material **[1]** and several later versions **[2, 3, 4]**. Other open sources were also studied to complete this general knowledge. These materials deal with comprehensive information about the aluminum electrolytic process. In order to suit the actual cases of cell operation, the common production problems are divided into six major groups. Figure 6.5 shows these six main groups and relevant detailed sub problems.

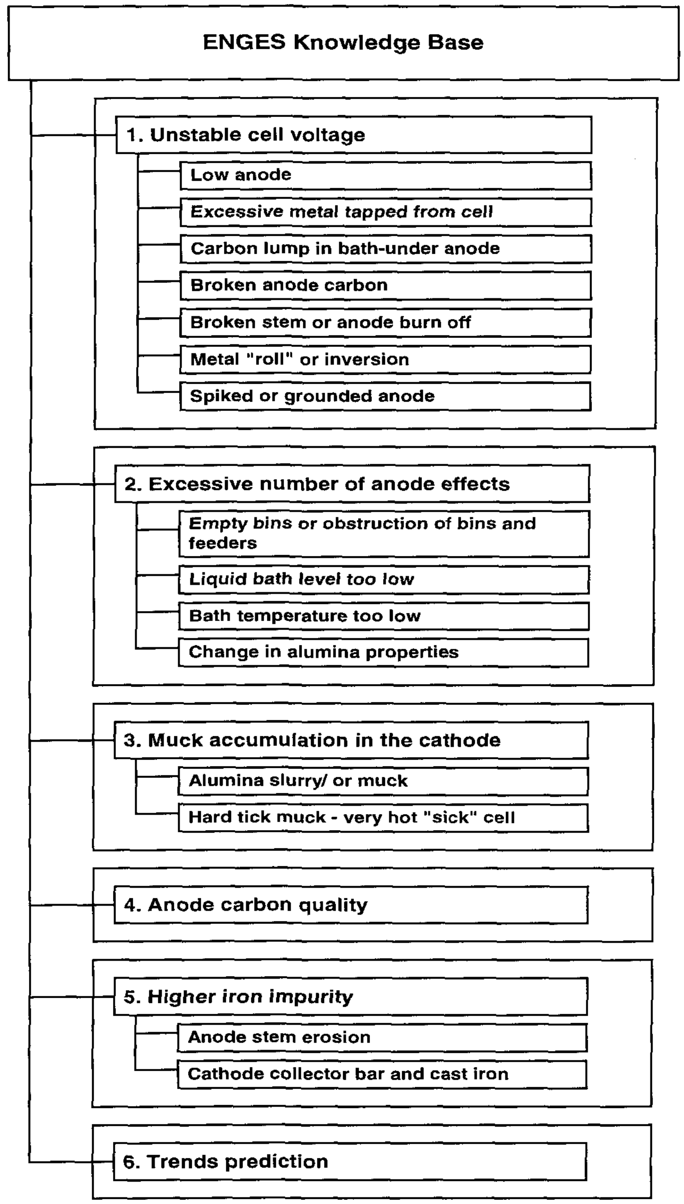

Figure 6.5 Main components of ENGES

6.4.2 Knowledge base of OPEES

Characteristics of the knowledge base of OPEES

The main difference between ENGES and OPEES is in the running mode. ENGES works only in the off-line mode, but OPEES can work in on-line mode as well. Therefore, the corresponding knowledge base is also different. The major OPEES characteristics are listed as follows:

- Applies to the particular type of cells
- Specific knowledge suitable for most types of cell
- Easy to compile
- Can be used as a training tool
- Accesses the real time process data
- Works as a consultant to help operator to solve the problems.

Main components of the knowledge base of OPEES

Based on the features of OPEES, the knowledge collected for the knowledge base building is mainly from the Horvath's report **[5]** and completed with other relevant sources. The knowledge of OPEES can be applied to four typical types of cells, the Söderberg anode or prebake anode; and low amperage and high amperage cells. In the knowledge base of OPEES, there are thirteen rule sets for the corresponding problems. Except these thirteen problems, the corresponding alarm modules are special for the on-line process data detection. It will be associated with all the process data of each type of cells. Table 6.3 lists these main components of the rule base of OPEES.

Table 6.3 Main components of OPEES

Main components of OPEES	
1	Average voltage of anode effect
2	Daily anode adjust number
3	Anode cathode distance
4	Anode effect number
5	Bath level
6	Bath ratio
7	Bath temperature
8	Cell voltage fluctuation
9	Iron impurity
10	Ledge profile
11	Metal level
12	Metal tapping
13	Silicon impurity

Another difference between OPEES and ENGES is that there is no problem selection part in the knowledge base of OPEES. All the user selection is based on the status of process data, which is detected by the alarm modules.

6.4.3 Modularization of knowledge base

One of the goals of our project is to design components that other expert systems or other system developers can reuse and that can easily extend to individual cells. This is why the knowledge base is modularized.

Modules are the building blocks of the knowledge base. This has the following advantages:

- Allows developers to divide and merge work
- Increases productivity, allowing developers to work on different pieces of the problem without interfering with other developers
- Results in potentially reusable modules
- Decreases debugging time and limits the impact of bug fixes and design changes
- Increases the flexibility and maintainability of the resulting system, because it is easy to upgrade or replace single modules independently of other modules.

6.4.3.1 Module types

Two types of modules are used for our knowledge base building. One is an independent module, which is the top-level module, in which definitions and instances of that definition are located in the same module. The other is a dependent module, which is the low-level module, in which an instance is located in the dependent module and its definition depends on the top-level module in the hierarchy.

There is only one type of independent module applied in the knowledge base of OPEES, which is the alarm module that specifies the process data alarm limitations and then converts the numerical data to the fuzzy expression. All other modules applied in OPEES are dependent modules, where the located definitions depend upon the conversions of alarm modules or other modules. They are the functional modules and structural modules, used for particular problem diagnosis purposes and logical organization between modules.

6.4.3.2 Module hierarchy of knowledge base

The knowledge base of OPEES consists of several module sets. They are organized around functional or structural boundaries, depending upon the needs of the application. The module hierarchy of the knowledge base of OPEES is shown in Figure 6.6.

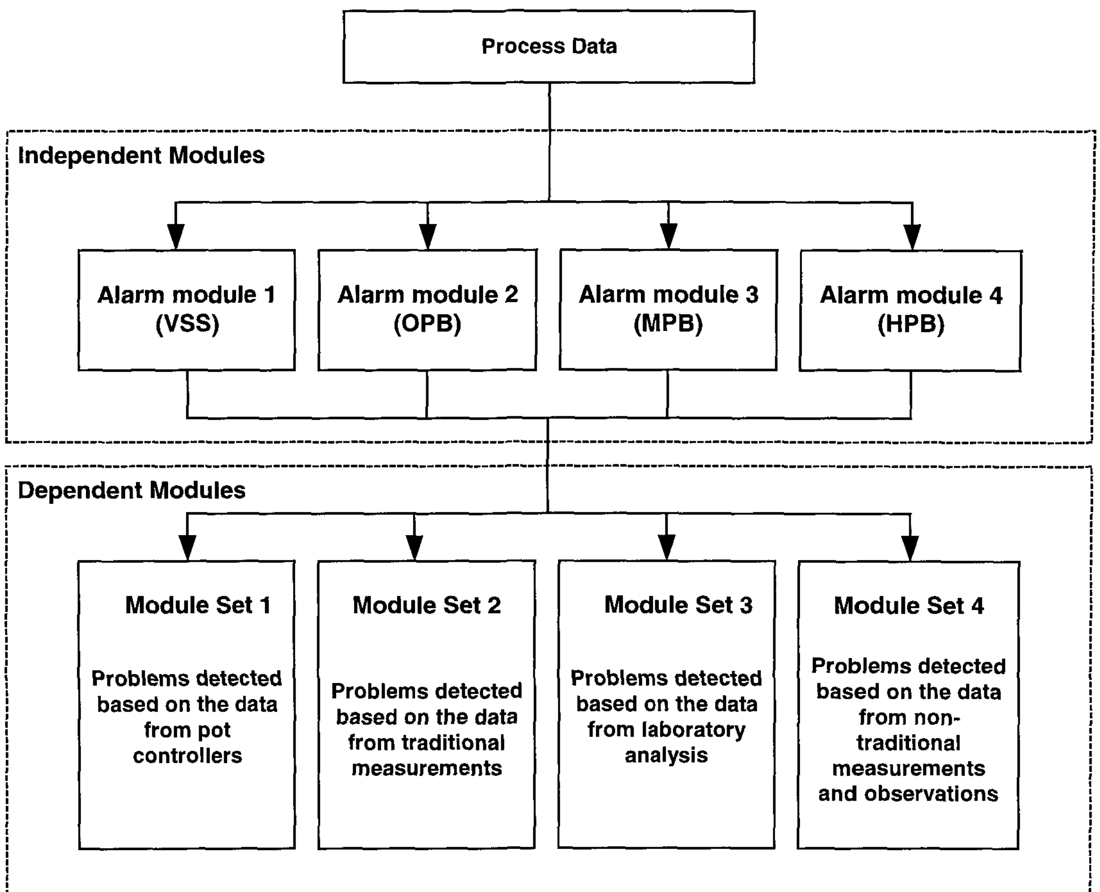

Figure 6.6 Module hierarchy of knowledge base of OPEES

Except for the alarm modules, all of the other twelve low-level modules of OPEES are dependent modules; they must request the necessary definition from other modules in the hierarchy. This means their work depends upon whether the reference came from the alarm modules or other modules. The direct process data treatment is only executed in the alarm modules; otherwise the low-level functional modules are not active. The functional modules only consider the conclusions of the alarm modules, where the numeral data is judged and converted to the fuzzy expressions. In order to reduce the restrictions of particular properties of knowledge, the low-level modules only treat the more general knowledge and do not need to be concerned with the particular numeral process data. If the expert system is required to work for other cells, the modification of the knowledge base is only necessary in the alarm modules.

Figure 6.7 is an example that shows how the alarm module converts the numeral value of bath temperature into fuzzy expressions.

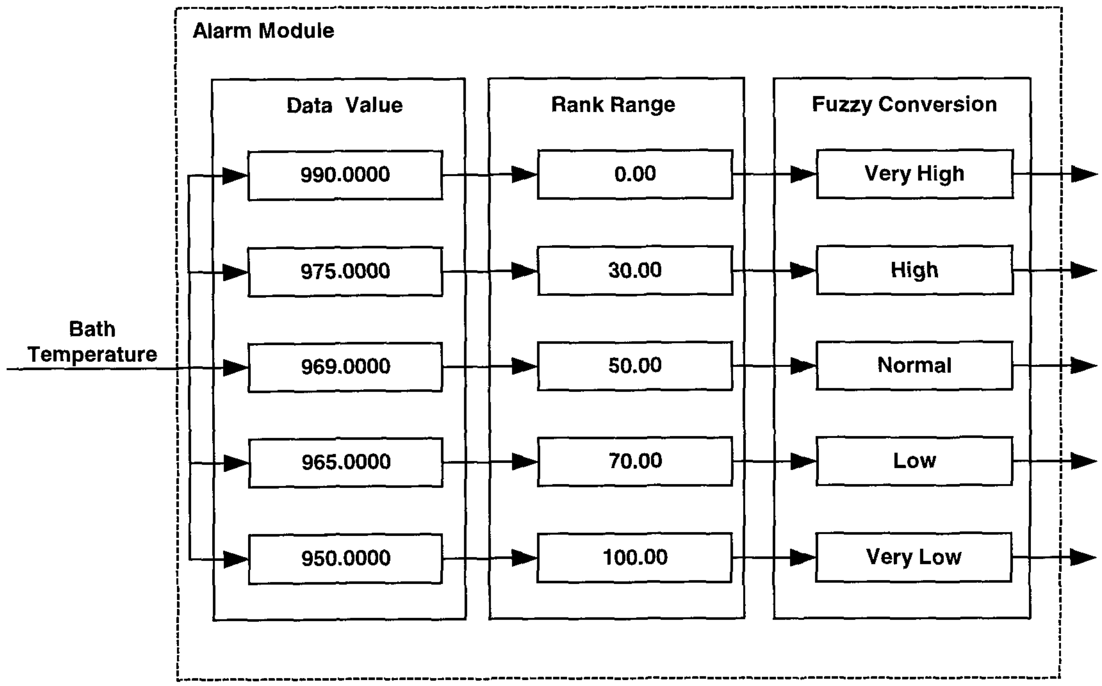

Figure 6.7 Example of alarm module

In this example, the bath temperature has been converted to fuzzy expressions "Very High," "High," "Normal," "Low," and "Very Low," which are often used by the operators in real cell production, especially for the non-continuously measured process data. All the relevantly functional modules use these expressions as their symptoms to do the reasoning. Such uniform fuzzy expression allows them to avoid direct connection to the particular process data; therefore it can be considered that the functional modules are only based on relative general knowledge. By this way, we can separate the knowledge into two parts: special knowledge and general knowledge. This organization will be of benefit to both the knowledge base design and further expert system extension.

6.4.4 Knowledge base maintenance

6.4.4.1 System maintainability

To maximize the maintainability of our expert system, the following steps are taken:

- Use modularity to increase maintainability

  By creating small, standard functional modules, we can define relevant "units" based on their functions for maintenance. Not only is it possible to deal with every aspect of the functionality of a small module, we also reduce the potential problem of bug fixes, design changes and system maintenance. When we use these modules as the element components, we can reduce the scope of our own maintenance efforts. This emphasizes the benefit of designing the application as multiple reusable modules, rather than as a single monolithic application.

- Build an user-accessible maintenance interface

  Considering the end user requirement, several specific interfaces are designed for expert system maintenance. These interfaces allow the user to modify the partial modules of knowledge base to suit the variations in cell production. Using these interfaces, the user can modify the process data alarm limitation depending on the cell situation. Of course the protection of the knowledge base must be considered, either for system running or system maintenance. Only the authorized users can obtain access to the modification of the knowledge base of the system.

6.4.4.2 Example of knowledge base maintenance – updating the process data alarm limitation

When an expert system needs to be extended to other cells or the operational condition has been changed during the production life, the corresponding knowledge must be modified to suit the new conditions. In the knowledge base of OPEES, the alarm modules are designed for communication with the particular process data of individual cells. They receive the on-line data and monitor their operating status. The alarm module is the sole module which directly connects with the particular information of individual cells. To meet the requirements of the particular cells and different situations, the alarm limitation must be variable. Therefore, it is necessary to provide corresponding environments that allow users to modify or to update the varied process data limitations. The interface "System Maintenance" is especially designed for this purpose.

Figure 6.8 is an example of process data alarm limitation value modification interface of OPEES-1. When the user's name and password have been verified, the authorized user can select the line number and cell number to confirm which cell should be maintained. In this example, the alarm limitation of cell voltage fluctuation is selected to do the modification. In order to simplify the user maintenance procedure, and to avoid the user's direct access to the knowledge base, special interfaces are designed.

By using these maintenance interfaces, there is no need to enter the specific limitation values, as only four values are required: "Normal top value," "Normal bottom value," "Maximum value," and "Minimum value." Then the alarm limitation is obtained by the programmed rules in the modules. The unit of the process data is also changed automatically depending upon the selected data.

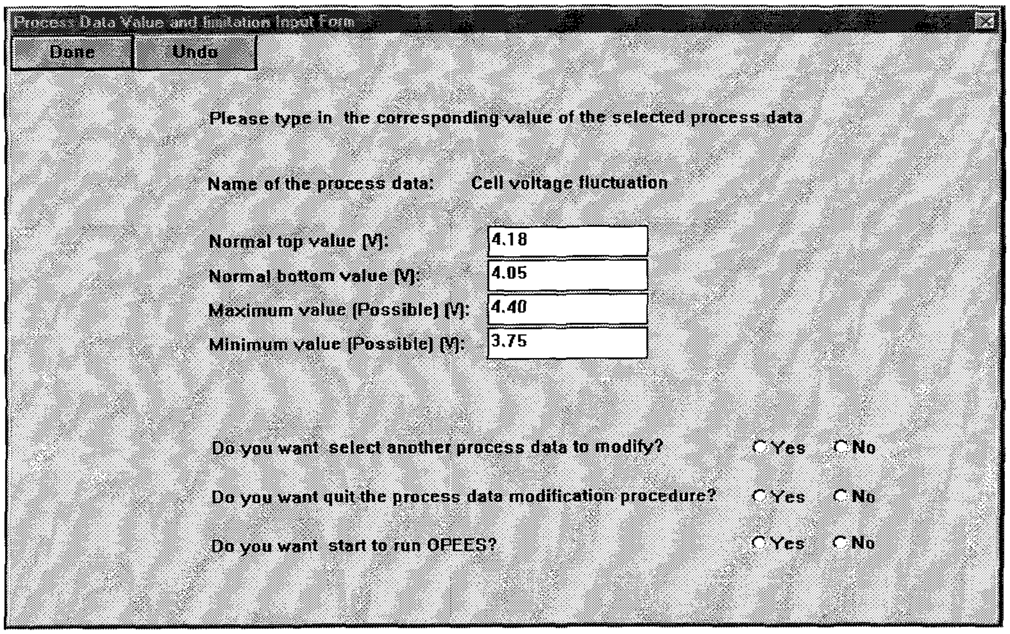

Figure 6.8 OPEES-1 alarm limitation value maintenance interface

Figure 6.9 illustrates an example of the logic structure of the determination of the cell voltage fluctuation alarm limitation, which is in the special module of OPEES-1. When the required value entry is finished, the alarm limitation values in the modules will be modified automatically. Then, the updating of the alarm limitation of the cell voltage fluctuation has been completed.

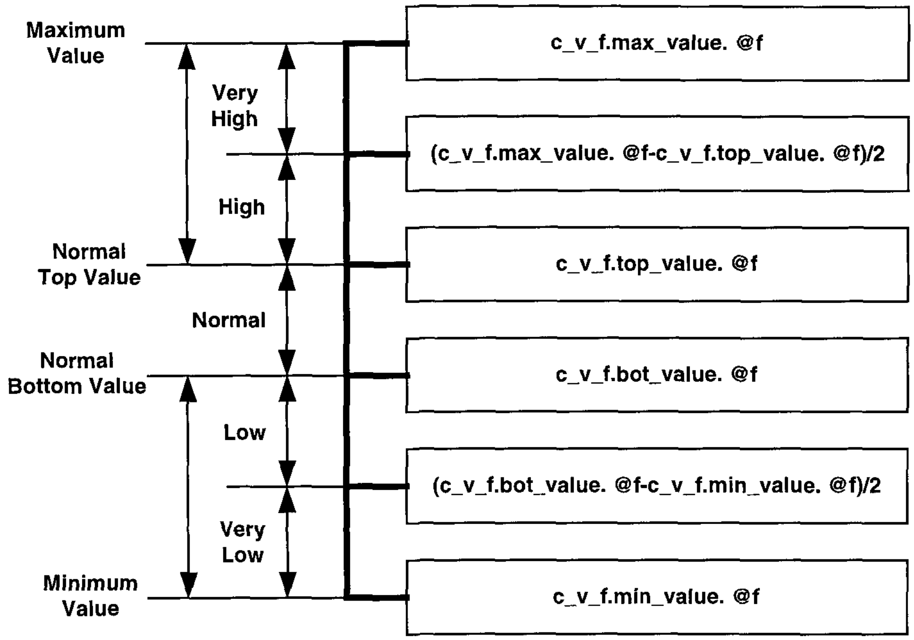

Figure 6.9 Example of determination of cell voltage fluctuation alarm limitation

Because different types of cells have different operational data, the alarm limitations of process data also differ. Therefore, four sets of alarm limitations are designed for each type of cells in OPEES.

6.5 Reasoning process improvement

6.5.1 Interface development

Interaction between an expert system and its user conducted in a natural language style was considered in the system design. The interface plays in a manner that is acceptable to the user, where the special demands come from the system. Therefore, the interface is an important component of the expert system; it was designed in parallel with the development of our knowledge base. The design and structure of the knowledge base is influenced by the design of the interface.

A basic design requirement of the interface is to ask questions; it is centered on the "ASK" function. In ENGES, it is the only way to obtain reliable information from the user. Therefore, attention needs to be paid to the design of the questions. Menus, graphics, or tailor-made screens are used in our interface designs.

There are two types of interfaces applied in our expert system development: user interface and development interface. The user interface is designed by the system developer. The development interface is intended for the knowledge engineer and is usually provided by the shell. The user interface can have simple textual displays or interactive graphics. The developer can develop the system using a source code approach or be led through various editors during system development.

The four types of user interface designed in our expert system are:

- System interface – The system interface is used for running the system, it contains two levels: introduction and intermediate interface.
- Question interface – The question interface is used to obtain information about the problem from the user. There are two different question interfaces for system-generated questions and system developer designed questions.
- Display interface - The purpose of a display interface is to present information to the user. There are two types of display interface typically used in our expert system: statement interface and graphical interface.
- Maintenance interface - This type of interface allows the user to modify specific items in the knowledge base. It would not interfere with the rest of the knowledge base.

6.5.1.1 System interface

During the running of the system, the introduction interface and the intermediate selecting interface will give the information about the relevant systems or procedures, and will allow the user to select the corresponding button to launch the relevant sub-system or procedure. These types of interfaces are structural interfaces associated with the relevant parts of the system that can decide the organization of the system. Figure 6.10 is an example of the introduction interface of AEPES, where three sub-systems are introduced and will load the corresponding system depending upon the user's selection. When the user chooses the desired sub-system and clicks the "Done" button, the selected system will start to run.

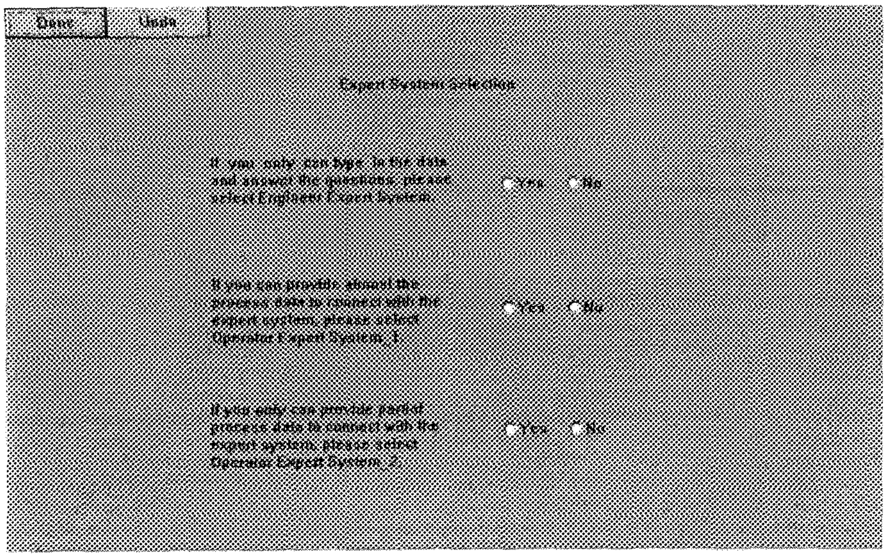

Figure 6.10 Example of introduction interface

6.5.1.2 Question interface

The question interface is the most often used interface; two types of question interfaces are applied to our system. One is the automatically generated question interface, the release of which is dependent upon the relevant rules. We designed another question interface for special purposes. The typical question interface has three basic parts: text portion containing the question, answer entry part, and control section. Figure 6.11 is an example of the system generated question interface.

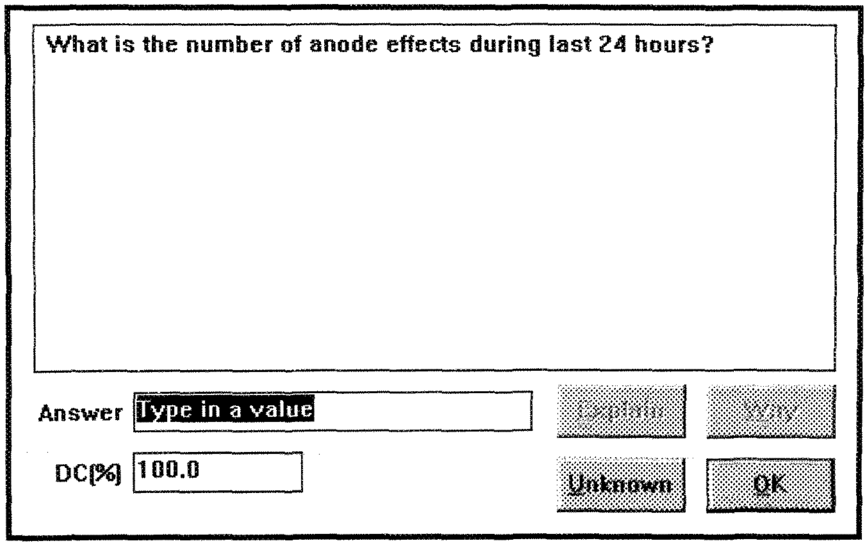

Figure 6.11 Example of question interface

Depending upon the question, the user may type in the numeral value or select "Unknown" or "OK" to answer the question. If the user does not know a certain answer, he can use the degree of certainty to describe how sure the answer is. All the information processes through the interface to the working memory, and then provides for the reasoning process.

6.5.1.3 Display interface

Most of the interface is text-based; the system asks questions and the user responds by either typing an answer or selecting from a menu of possible answers. But the graphical interface gives the user more impressions, especially for some complex cases, that are difficult to describe by text. We designed several graphical interfaces for both questions and final suggestions. Figure 6.12 is an example of graphical interface, where the final suggestion is provided by both the text and the graphic. The case of "Too close A-C distance" is described in the graphic, where the definition of A-C distance and the problem occurred is illustrated. The suggestions describe the detailed action steps in text. This allows the user to better understand the problem and how to solve the problem.

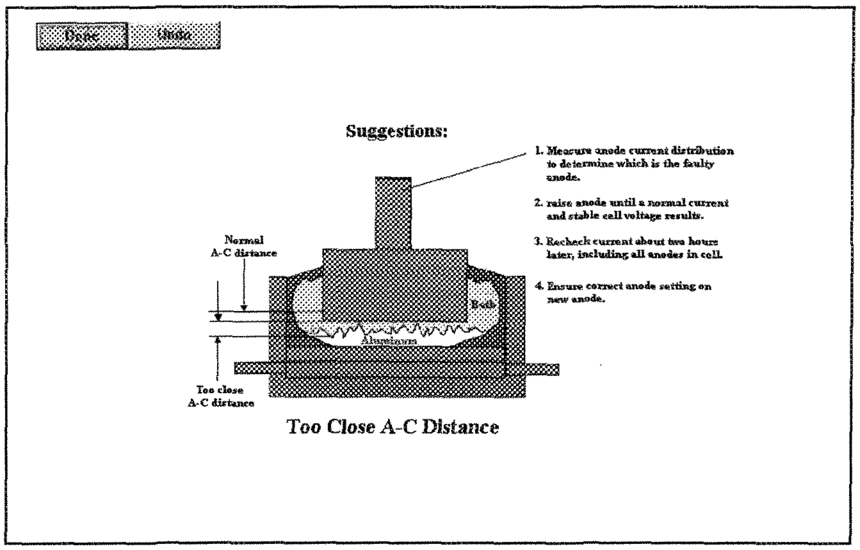

Figure 6.12 Example of graphical interface

6.5.1.4 Maintenance interface

Compared to other types of interfaces, the maintenance interface needs more attention to design. The major reason is that the maintenance action will require direct access to the knowledge base. The corresponding protection must be considered, such as the scope of modification and the determination of the authority of the person doing the maintenance. The detailed description and an example can be found in **Section 6.4.4 "Knowledge base maintenance."** This type of interface allows the user to modify or to update the specific information of the knowledge base. It is important that the modification would not damage the knowledge base.

6.5.2 Uncertainty technique application

In practice, we often have to deal with some information which is uncertain, vague, incomplete or inconsistent. All these imperfections are different in nature and lead to different problems. Therefore, the human experts must make judgments when solving a problem while the information on the problem may be suspect or incomplete, and some of the knowledge for interpreting the information may be unreliable. These difficulties are also found in expert systems. In many cases, we can find that the user's answers for the required questions are not completely true or false, but are in an intermediate state. This situation leads to a search for techniques to manage inexact reasoning, which results in a need to develop systems that can draw conclusions under uncertainty, vagueness or incomplete information.

6.5.2.1 Certainty factor

One of the probability theories, which are commonly applied for exact reasoning in expert systems, is called certainty theory. This theory grew out of the work on MYCIN, which offers a practical technique for performing inexact reasoning in many expert system applications. It depends on the judgmental belief values given to the uncertain statement and is suited for problems that lack a strong statistical basis.

To develop the certainty technique, two important concepts are considered: representing the unknown (how much is known and how much is unknown) and the reasoning unknown (when draw the conclusion based on know/unknown). Therefore, the following different information would be considered for the certainty techniques:

- Unreliable data: faulty sensors, defective measuring instruments, biased referees, and factors not taken into consideration, such as changing conditions
- Incomplete data: complete information not always available, especially for the cells
- Imprecise data: approximative measurements, manual measurements, limited number of samples.

The same types of uncertainty can also be present in the rules for drawing conclusions from the data, where the requirements for reasoning with uncertainty are concerned with the representation of uncertainty and the combination of uncertain pieces of information. No general agreement exists on what is the best method. Generally, two basic theories about certainty are considered: probability theory and certainty factor model. The former has a good theoretical foundation, but has problems with applicability whereas the latter is easy to apply, but is difficult to justify theoretically.

Certainty factors are subjective, expert-based belief measurements. They are used in some systems to indicate the strength of the given evidence. In our system the certainty factor (CF) is used to treat the uncertainty information. The CF was developed using a simple model specifically for use with the rule-based expert systems, which are often used in practice **[9, 10]**.

The basic feature of certainty theory is how uncertainty is conceptualized and manipulated. The starting point for any method of uncertainty theory for inexact inference is a method for expressing the degree to which each fact or rule is true. Then, a method for propagating uncertainty is needed as rules are applied during the inference process. The CF is a number that reflects the net level of belief in a hypothesis given the available information. The CF, as in fuzzy logic, assumes that each statement has a degree of truth, which is a number between 0 and 100. Absolute truth corresponds to 100, while certainly false is represented by 0.

In the expert system, the user assigns the CFs to the facts by question-responses. The minimum CF of the facts becomes the CF of the solution, unless multiple rules arrive at the same solution. In such cases, the certainty of the solution is calculated according to the formula A + B - A*B/100, where A and B are the CF of solutions reached by separate rules. To reason with CFs, the system needs to be able to calculate the degrees of confidence for statements connected by ANDs and ORs. We write CF (A) for the certainty factor of A. The formula we use to calculate CFs for conjunctions (i.e., ANDs) assumes that the chain is as strong as its weakest link and therefore takes the minimal CF in a conjunction. On the other hand, in a disjunction (i.e., ORs) we have a choice between two propositions and may rely on the strongest level of confidence. We therefore use the maximal CF in disjunction. Formally, this means that **[11]**:

$$CF\ (A\ AND\ B) = minimum\ [CF\ (A),\ CF\ (B)] \qquad (6\text{-}1)$$
$$CF\ (A\ OR\ B) = maximum\ [CF\ (A),\ CF\ (B)] \qquad (6\text{-}2)$$

To calculate the final general CF, it is necessary to combine the CFs of each piece. For calculation of the combined CFs, the product method is the prime need:

$$CF = (combined\ conditions\ CF * conclusion\ CF)\ /\ 100 \qquad (6\text{-}3)$$

Then, the summary CF is calculated as:

$$CF = (combined\ conditions\ CF + conclusion\ CF) - (combined\ conditions\ CF * conclusion\ CF)/100 \qquad (6\text{-}5)$$

6.5.2.2 Certainty technique application

There are two concepts about the certainty techniques applied in our work: certainty factor (CF) and degree of certainty (DC). While the basic theory is identical, the former is used in the conclusion statement of rules; the latter is used for the question interface.

The CF is used to present the belief assigned to uncertain information. The 0 to 100 scale is used where the numbers are given a range of 0 (definitely false) to 100 (definitely true). A positive value represents a degree of belief, while a negative value indicates a degree of disbelief. For example, if the operator states that "anode position high" was probably true, a CF value of 65% is assigned to this case.

During consultation, a keyword triplet may already be instantiated, yet through assignment, database access or other events, the system may attempt to assign a new degree of certainty to the triplet. By default, Comdale/X maintains the lowest degree of certainty. We can customize the method of accumulation of certainty by using the accumulation facet of a keyword triplet. When the system starts the reasoning, each conclusion statement in a rule of our system is accompanied by a CF. The value of the CF indicates the confidence in that conclusion statement, if the premise of the rule were 100% true.

The DC uses the same numerical value with ranges also from 0 to 100; it is a measure of how sure the system is that the value of attribute is true. The DC reflects uncertainty in data. During the inference, the DC is used in association with the key words: “Not Known,” “Known,” or “Unknown.” “Not known” is used before inference commences. “Known” is used when it is assigned a DC. “Unknown” is used when the system tried unsuccessfully to make the triplet “Known.” Here is an example rule used in ENGES, which is associated with CF:

IF *anode_effect.number.high*
AND *crust.type.very_hard*
THEN *bath.temperature.too_low* ***is TRUE*** *cf=90*
THEN TEXT *“The bath temperature is lower than normal! Go further to check other relative reason.”*

It indicates that if we are 100% sure that the anode effect number is high and crust is very hard, we are only 90% sure that the bath temperature is too low.

However, we found in many reasoning cases, that the system would often need to work with uncertain or even unknown information. For instance, the user often analyzes the available information using qualitative terms or phrases such as “probably,” “it is likely that ...”, “it almost seems certain that ...”. Therefore, how to handle the CFs are a very important and a difficult task of system design, and more detailed knowledge is often needed. If we cannot handle the certainty factor correctly, a completely different result will be obtained.

6.5.3 Multiple symptoms criterion – example of certainty theory application

In the rules of the knowledge base, the symptoms are used to describe the phenomena of the problems that occur. Normally, most of the problems have multiple symptoms.

In the knowledge we used, we found that all symptoms of the corresponding problem are only listed as such; there is no specification about the rank of the symptoms. Generally, the rules use IF-THEN and IF-THEN-ELSE structures in a rule based system, where the symptoms are treated as the premise in the IF part of the rules.

In the following rule, found as an example in a previous system, all the symptoms are arranged by using logic AND:

IF	*instability.factor.high*		
AND	*anode.current.high*		
AND	*anode.currents.unstable*		
AND	*anode.burn_off.detected*		
AND	*anode.stem.red*		
AND	*anode.spike.detected*		
THEN	*anode.position.low*	***is***	***TRUE***

This rule dictates that the keyword triplet "anode.position.low" will be set to be TRUE whenever multiple conditions are to be satisfied in the premises. By virtue of Equation (6-1), the general CF of all the terms is equal to the minimum one among all CFs. If any one premise is not satisfied, that means this premise is definitely false or has a very low value of CF. Therefore the general CF also has to be 0 or a very low value of CF. The conclusion in the THEN part of this rule is consequently set to be FALSE. During the inference process, if some error occurs with any premise, even in a minor premise, that error will influence the correction of the inference result. For instance, consider the case where all the premises are considered to be TRUE in confidence, except only the one error message that comes from the imprecise manual measurement of spike, as the operator does not find the spike under the anode. This incorrect information will let the keyword triplet "anode.spike.detected" set to be FALSE. Its CF becomes a very low value, which leads the general CF of conjunctions to the same low value. Figure 6.13 provides an illustration of this multiple symptoms working procedure.

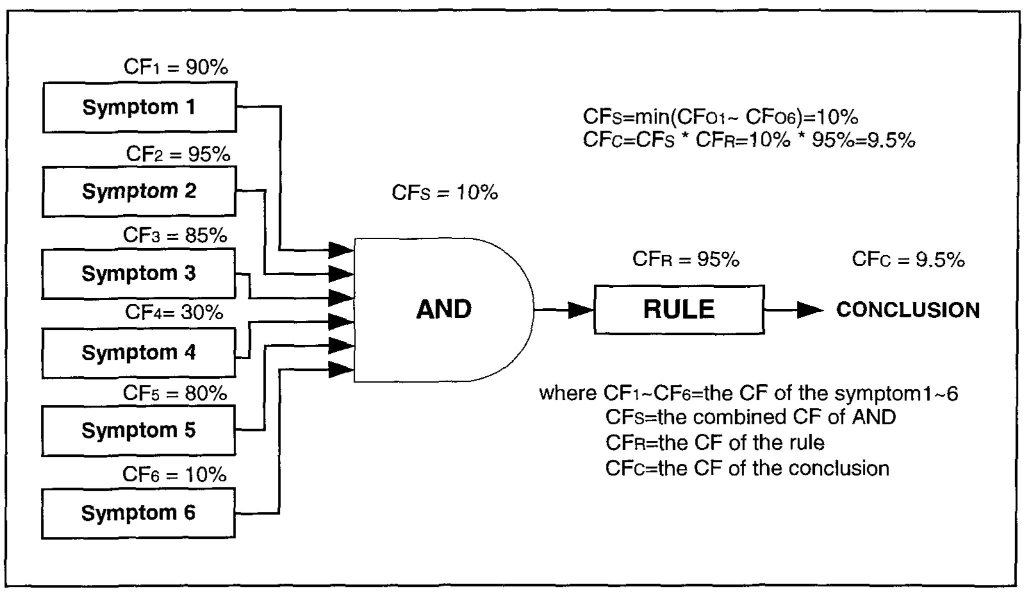

Figure 6.13 Example of multiple symptoms working procedure

To avoid such a situation, shown in this particular case study, the importance of each symptom should be qualified. For the individual case, the importance of the relevant symptoms is different, and the correct result of the inference process strongly depends on the most important symptoms. Otherwise we should adopt some steps to avoid the influence of an error message from the less important symptoms. Therefore, we focus on improvements of the description of the symptoms using a quantification method, which depends upon the corresponding quantitative description of their importance. Then, we can calculate the general CF by combining such quantification of all symptoms. When this CF reaches the presupposed value, the conclusion is considered to be TRUE. Using this method the error of a minor symptom will not be able to influence the final inference result.

We have discussed this topic with the domain experts, they point out that it is impossible to indicate a constant quantitative description of importance to each symptom. Because the real cell is a dynamic process, for different cells the symptoms vary with the different cases. Therefore the importance of the symptoms varies also. For example, in the case of "anode position is low," sometimes the symptom "anode burns off" does not appear. Or, the anode stem does not become too red. All these symptoms will appear in different levels, or may not happen at all. They strongly depend on the particular states of the cells.

Using fixed quantification to describe the symptoms may not suit the variation of dynamic process, however the variations of the value of quantification could still be assigned within a particular range. Therefore, we focus on how to express the corresponding ranges. Based on the proposal from the domain experts, the quantification of the symptoms of certain problem may be described by two sets of expressions: major symptoms and minor symptoms. For different problems, the multiple symptoms of each set can be divided into several levels to describe the different levels of the symptoms. Applying the composite CF calculation and logical ANDs and ORs, the previous example will be reorganized as follows:

IF *instability.factor.high*
AND *anode.currents.high* / *anode.currents.unstable*
AND *anode.spike.detected* / *anode.stem.red* /
anode.burn_off.detected
THEN *anode.position.low* ***is TRUE***

The arrangement of the different levels of symptoms can be found in Figure 6.14, where we use a logical AND to combine three symptom sets. The three major symptoms are divided into two levels; the logical OR is used to combine the two symptoms of level 2. Similarly, the three minor symptoms are also combined by the logical OR.

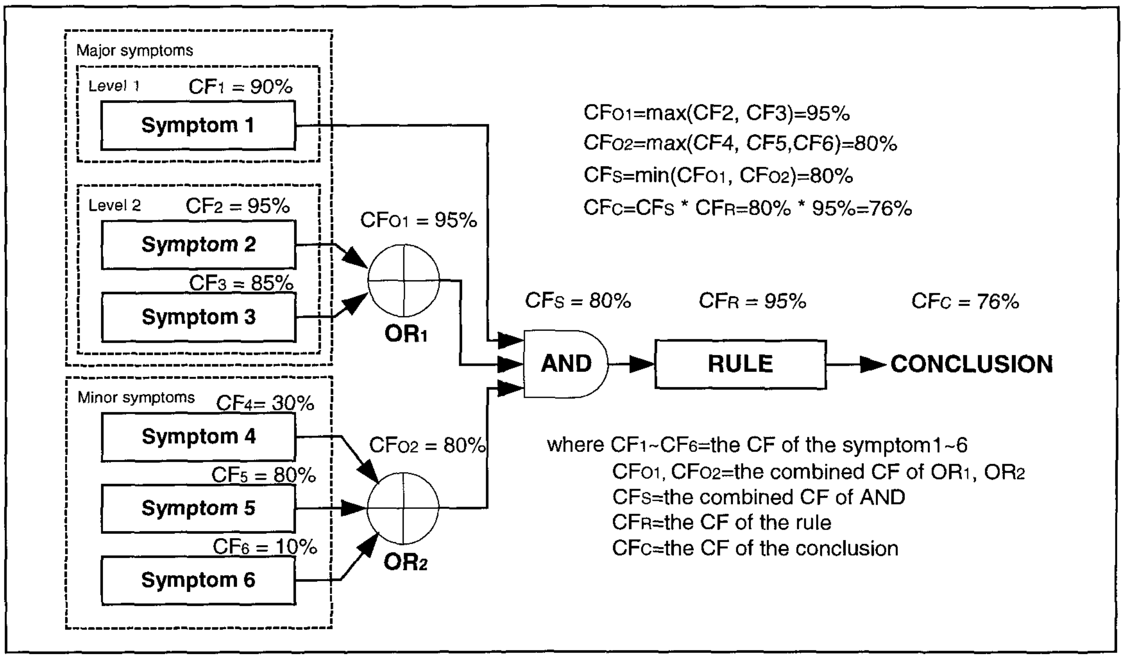

Figure 6.14 Improved working procedure of the example rule

Through this procedure, the influence of an error message on an individual minor symptom has been reduced. Consequently some unreasonable results can be avoided and the reliability of reasoning process is improved.

6.6 Example of ENGES

Here is a typical example of ENGES - diagnosis of the unstable cell voltage. Depending on the problem that has been selected, the corresponding sub-system will be loaded. As ENGES works in off-line mode, all the necessary information about the problem will be gathered by the question interfaces. An example of question interface is given in Figure 6.15, where six typical cases of cell voltage deviation are provided. The user is asked to select one close to the measurement.

The inference engine is the processor of the knowledge; the inference process will be invoked once the desired problem is selected. The reasoning process will be stopped when all the facts connected in chaining have been searched. The inference engine checks the facts matched and considers the relevant knowledge to make a conclusion.

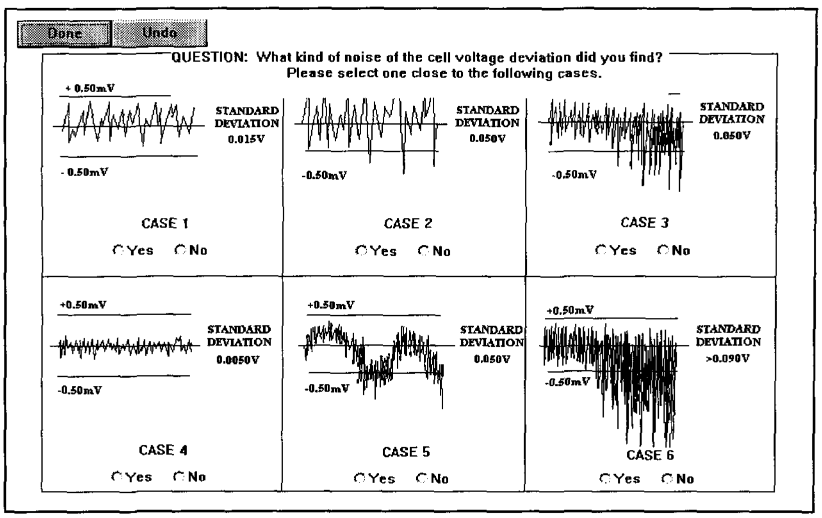

Figure 6.15 Question interface for unstable cell voltage cases

Here is an example of rules, which are in a chaining connection, the conclusion generated by the rule *"broken_stem2"* by "THEN" will send the corresponding triplet to the "IF" of following rule *"broken_stem3"*. Figure 6.16 shows the connection between two related rules and Table 6.4 lists the detailed contents of these two rules.

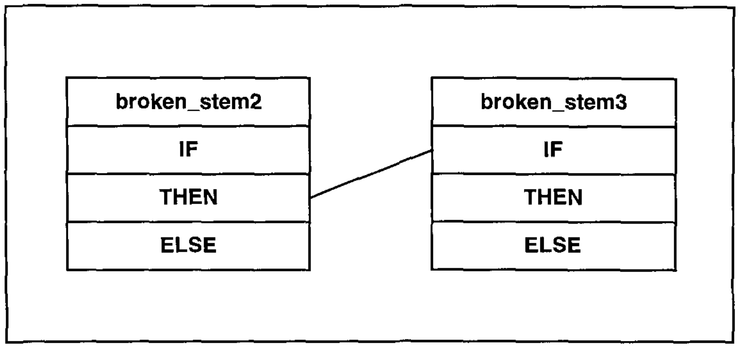

Figure 6.16 Chaining connection between example rules

Table 6.4 Rule "broken_stem2" and "broken_stem3"

Rule   ***@ name=broken_stem2***
IF ucv.case3.occurred   *AND anode.current_individual.low*   *AND anode.current_individual.unstable*   *THEN anode.current.abnormal is TRUE*   *end Rule*

Rule   ***@ name= broken_stem3***
IF ucv.case3.occurred   *AND anode.currentabnormal*   *AND anode.current_all_other.higher*   *THEN anode.broken.stem is TRUE*   *THEN FORM ("F11.frm")*   *endRule*

Depending upon all of the necessary information being gathered, the inference engine follows the search strategy to find the problem and displays the diagnosis result in Figure 6.17 with the illustrated suggestion be found in Figure 6.18.

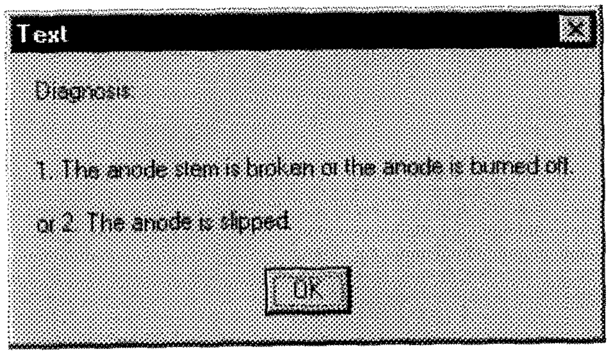

Figure 6.17 Example of diagnosis result

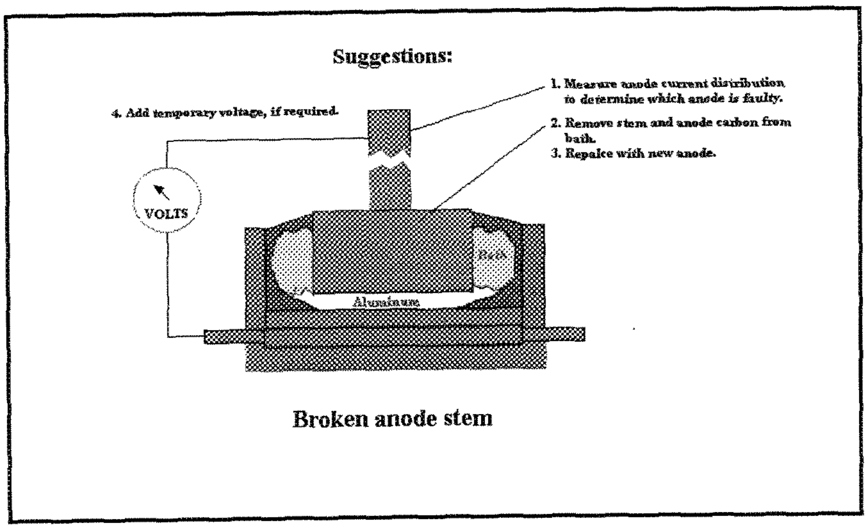

Figure 6.18 Example of illustrated suggestion

6.7 An analysis of the results

As the motivation behind this project is to design an expert system for a practical use, the emphasis is on knowledge base programming, interface design and system maintenance. Based on the test results, we can give the following comments about these aspects.

Knowledge base programming

To improve the flexibility of knowledge base programming, the use of a modular structure is an efficient step. For OPEES, the alarm module and other functionality modules are separated. The alarm module is designed only for connecting with the particular process data of individual cells, but the general knowledge is coded into other modules. Therefore, for any particular cell or for different cases, modification work is only concerned with the alarm module and this will help the user in making the module suitable for different cells.

Interface design

The interface is the most important link between system and user. We designed the interface for two purposes: to gather the necessary information from the user and to send the user a clear and accurate message. For the latter purpose, the message can give the user suggestions to solve the problem. But for the former purpose, because of the limited capacity of the Comdale shell, only user selection and entry of the values are available. The system cannot carry out the interactive communication in the real sense, therefore cannot feedback the opinion of the user through the interface, which would be useful for the reasoning process.

System maintenance

In OPEES, the alarm limitations are designed to be modifiable for the different cases. Although this is only a limited maintenance capacity, it is efficient for application to different cells.

CHAPTER 7 AEPES – ADVANCED DESIGN

7.1 Objective of advanced design

While the expert system was under development using Comdale/X, a search was underway for a new shell in order to improve the performance of AEPES. This was mainly motivated by the limited on-line capacity of Comdale/X. The new shell should have enhanced performance in system design, interface edition and powerful communication ability. Based on the ability to inherit from the advantages of Comdale/X, its new generation shell, SmartWorX Suite, was our first choice. Unfortunately, the formal version of SmartWorX Suite suffered from last-minute unavailability due to Comdale Technologies (Canada) Inc. declaring bankruptcy in late 1998. Faced with this situation, we had to search for another available shell to continue our project. The new shell selected for the expert system development was Gensym's G2.

To improve the performance of AEPES by using G2, the major tasks are: Make a migration of the knowledge base from the Comdale version to the G2 version, then redesign a new structure of the system and select a reasonable communication bridge to realize the on-line mode.

7.2 Basics of G2

Gensym Corp. is the major company in the AI market. The G2 is the main product among their expert systems. Since the company began in 1986, there have been more than 13,000 product licenses applied in communications, manufacturing, aerospace, transportation, government and other industries. Gensym Corp. develops expert operations software that model, simulate and monitor production or business processes. In this work we apply the new version 5.1 of G2, which has gone through major upgrades in user interfaces and connectivity to host systems.

An introduction, brief characteristics and application examples of Gensym G2 are given in **APPENDIX 1.2 "Gensym G2"** and **APPENDIX 1.3 "Application examples study."**

7.3 Construction of system

The required knowledge of the advanced system is identical to the previous system. The system uses the same name: AEPES, which also consists of two sub-systems: ENGES and OPEES. However, the system structure is different, where the modules organization, the rules coding, the individual knowledge base, the interface design, and communicative tool are done in different ways. All these changes are based on the features of the G2 core and corresponding communication bridge [1].

7.3.1 Modularized system structure

Although we have organized the knowledge base by the modularized rule sets in the Comdale version, G2 provides a more systematically modularized knowledge base building utility that allows building and maintaining the system to be more flexible. The modules can be used to form a module hierarchy, which specifies the hierarchical dependencies between modules. Using such standards of the module architecture, a modularized knowledge base is constructed.

7.3.1.1 Module organization

Depending upon the requirement of the present work, two types of modules are used: the structural module and the functional module. The structural modules are used to organize the module hierarchy, but the functional modules are concerned with the reusable modules and toolkits that are used to realize the different functions.

The structural modules contain the classes and sub classes, which organize the common characteristics and behavior of similar objects; they are created only for the present system. The functional modules are mainly provided from standard G2 utilities, they provide G2 standard modules as the blocks of module hierarchy. For instance, the G2 User Interface Development Environment (GUIDE) is used to create graphical interfaces, which are automatically organized into the system module hierarchy.

In order to simplify the system, the same names of classes and sub classes are used to name the corresponding modules. Table 7.1 shows the module configurations of the present system:

Table 7.1 Modules of AEPES

Top-level module	Lower-level 1 module	Lower-level 2 module
AEPES	ENGES	ENGES-1 (Unstable cell voltage) ENGES-2 (Excessive number of anode effect) ENGES-3 (Muck accumulation) ENGES-4 (Anode carbon quality) ENGES-5 (Higher iron impurity) ENGES-6 (Trends prediction)
	OPEES-1	OPEES-1-1 (VSS) OPEES-1-2 (OPB) OPEES-1-3 (MPB) OPEES-1-4 (HPB)
	OPEES-2	OPEES-1-1 (VSS) OPEES-1-2 (OPB) OPEES-1-3 (MPB) OPEES-1-4 (HPB)

To create a hierarchy modularized system, the first step is to set up a top-level module, which is the root of module hierarchy. When the AEPES is set as the top-level module, it directly requires three lower-level modules: ENGES, OPEES-1 and OPEES-2. The ENGES-1.1 to 1.6, OPEES-1.1 to 1.4 and OPEES-2.1 to 2.4

are then directly required for ENGES, OPEES-1 and OPEES-2 respectively. After the structural modules were created, it was needed to specify the relevant standard functional modules, and then the whole module hierarchy was set up. Figure 7.1 shows the screen copy of the partial module hierarchy of AEPES.

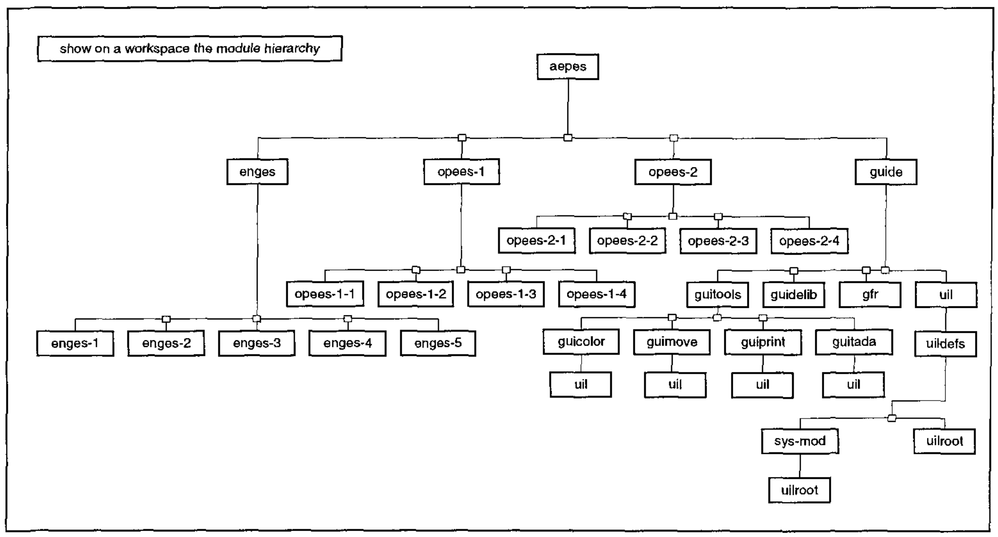

Figure 7.1 AEPES module hierarchy

7.3.1.2 Advantages of this system

With such a modularized structure, several advantages were obtained for the system building:

- The modularized technique facilitated the system development. The development of the ENGES started with few essential modules, which only contain the structural module "Unstable cell voltage" and necessary functional modules. When the test run was successful, the additional modules, which were developed separately, merged into the main system. This allowed a faster set up of a small, basic, but complete system that was easier to verify.

- The system can easily be expanded by merging the modules. For example, to increase the additional diagnosis problem "Trends prediction" to ENGES, only two steps are needed: (1). Create the ENGES module, which contains the file about the corresponding sub knowledge base. (2). Specialize the corresponding top-level module and the directly required lower-level modules. Then this new module is merged into the module hierarchy.

- Each module developed can work separately, without interfering with other module developments. The system can be divided into smaller modules, which are easier to handle, maintain, upgrade and replace.

7.3.2 Rule statement

The heart of any expert system is the ability to reason about the knowledge it contains. A rule is a special kind of statement that reasons under the given conditions and draws conclusions. The major difference in the rule editor between Comdale and G2 is their editor language. The syntax is used to represent a rule in the ASCII based file in Comdale, but G2 supports the natural language text editor to edit the proper syntax to construct rules.

Natural language statement

There are two major features of G2's natural language, which allow the rule creation to become easier and more flexible. The first feature helps by using human natural language liked syntax to edit rules and similar text expressions, such as methods and procedures, which is an easier way to do the text editing and reading. During the coding process, there will be offered some groups of precoded words for selection. Regardless of the antecedent or the consequent of the rules, when the primary word is selected, the corresponding words are released for further word selection. By this method, we can use natural language to do the editing as desired, but we must adhere exactly to the G2's format.

The second feature allows for reducing some limitations of the rule edition, which are often found in other expert system software editors. For example, Comdale only provides the IF-THEN-ELSE format to present the knowledge and to handle the relationships between the facts. However, for complex processes, more expressions are needed to describe the different cases. If there is an event concerning a fact, the presentation of the fact is not facilitated by using the IF rule format. However, depending upon the different purposes of the rules to be performed, the natural language offers five basic kinds of rules for the inferencing process: IF rules, INITIAL rules, UNCONDITIONAL rules, WHEN rules, and WHENEVER rules.

Figure 7.2 and Table 7.2 show an example of the difference between the two rules inferencing processes and syntaxes. Normally, the IF rule is used for data-driven processing, and the WHENEVER rule is used for event detection. In this example, for the given case, which was concerned with the event of cell voltage fluctuation detection, using the IF rule, the forward chaining was applied for detecting the data changes in this event. Forward chaining represents a powerful way to detect the variation of concerned data. But for a complex process, there are more IF rules that need to follow the complex path of the execution that will be inconvenient to debug, test and maintain. Thus, one should consider using the WHENEVER rule to detect this kind of complex event, which is concerned with every relevant data and phenomena.

In the present system, the WHENEVER rule and IF rule are two basic types of rules for rule-based processing. The WHENEVER rule is used for general event detection, whereas the IF rule is used for event detection based data-driven processing. If WHENEVER rules are used to detect an event, they must invoke some type of sequential processing, such as a method or a procedure.

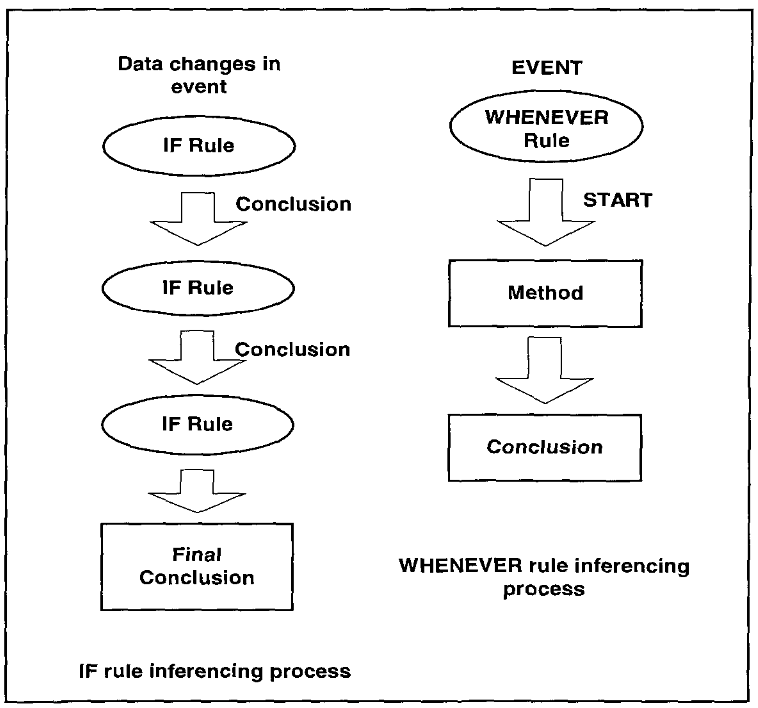

Figure 7.2 Illustration of two kinds of rules processing

Table 7.2 Rule syntax comparisons between IF rules and WHENEVER rules

Comdale Rules	G2 Rule
IF *cell.voltage_deviation.@float > 0.05000V* ***AND*** *cell.voltage_deviation.@float <= 0.120000V* ***THEN*** *cell.voltage_deviation.@string* ***is "HIGH"*** ***IF*** *cell.voltage_deviation.@string* ***is "HIGH"*** ***AND*** *anode.current_individual.high* ***is TRUE*** *AND anode.current_individual.more_unstabl* ***is TRUE*** ***THEN*** *anode.position.abnormal* ***is TRUE*** ***IF*** *cell.voltage_deviation.@string* ***is*** *"HIGH"* *AND anode.position.abnormal* ***is TRUE*** ***AND*** *anode.spike.occurred / anode.red_stem.occurred / anode.burn_off is TRUE* ***THEN*** *anode.position.low* ***is TRUE***	***Whenever*** *the deviation of cell voltage is higher than 0.050v,* ***start*** *low-anode () to diagnose the anode position.*

Another example of rules applied in this version is shown on Figure 7.3, where both the WHENEVER rule and IF rule formats are applied for iron impurity detection.

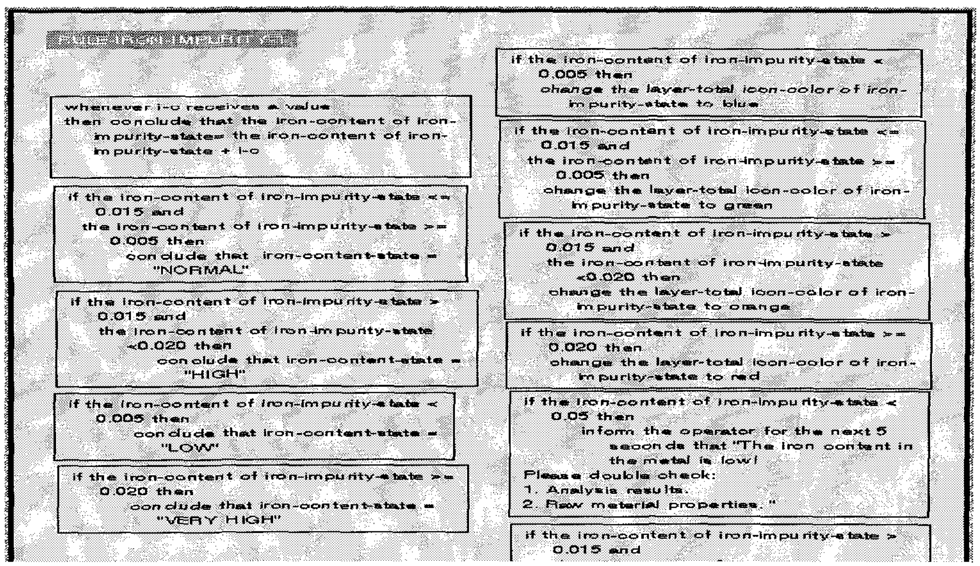

Figure 7.3 Examples of rule statement

The use of natural language rule editor shows several important advantages for the system development:

- More types of rule editor provide more choices to edit the different types of rules to suit complex cases.
- Natural language similar to human language has greater facilities to create various text expressions.
- Using the WHENEVER rule for event detection will simplify the rule structure and the rules are easier to test, trace and maintain. And the reasoning process will be faster based on the simpler rule structure.
- With the natural language editor, the coding of rules, methods, procedures and the inferencing process can be more efficient than Comdale and other similar expert system shells.

7.3.3 Interface design

Most of the user interfaces applied for AEPES are designed by using G2 GUIDE (Graphical User Interface Development Environment) and GXL (G2 XL Spreadsheet), which allow us to design the graphical dialogues and the spreadsheets for the present system. Using these interfaces enables the end users to view the system's messages and edit the information to realize the communication between the user and the system **[2, 3]**.

Depending upon the structure of the system, four types of user interfaces could be used:

- System interface
- Question interface
- Display interface
- Maintenance interface.

7.3.3.1 General interface configuration

As the structures of the two versions of AEPES are similar, the user interfaces also follow the same pattern to manage their sub-systems. Figure 7.4 gives the general configuration of user interfaces.

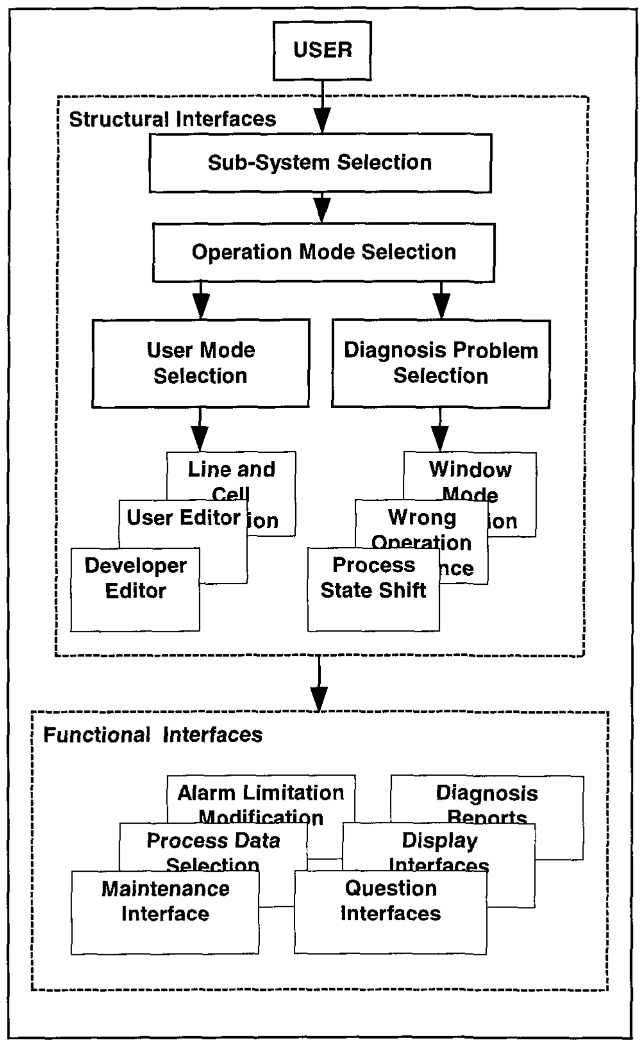

Figure 7.4 General user interface configuration

<u>Characteristics of interface</u>

Although the forms of user interfaces used in G2 version are similar to those used in the Comdale version, some operating functions are different. The major differences are:

- The interfaces are developed in a modular fashion. Therefore, they can be integrated into the system's modular hierarchy, and this increases the reliability of the system design.
- Depending upon the features of the interface developing tools, more facilitated user interfaces can be created, which allow users to perform the corresponding tasks more flexibility.

7.3.3.2 Example of system interface navigation

In Figure 7.5, the activation of the system interfaces' navigation shows the procedure that uses the navigation button function, which is supported by the utility modules of G2 GUIDE (Graphical User Interface Development Environment), to organize the system interfaces, which gives users more freedom to manage the system's running status.

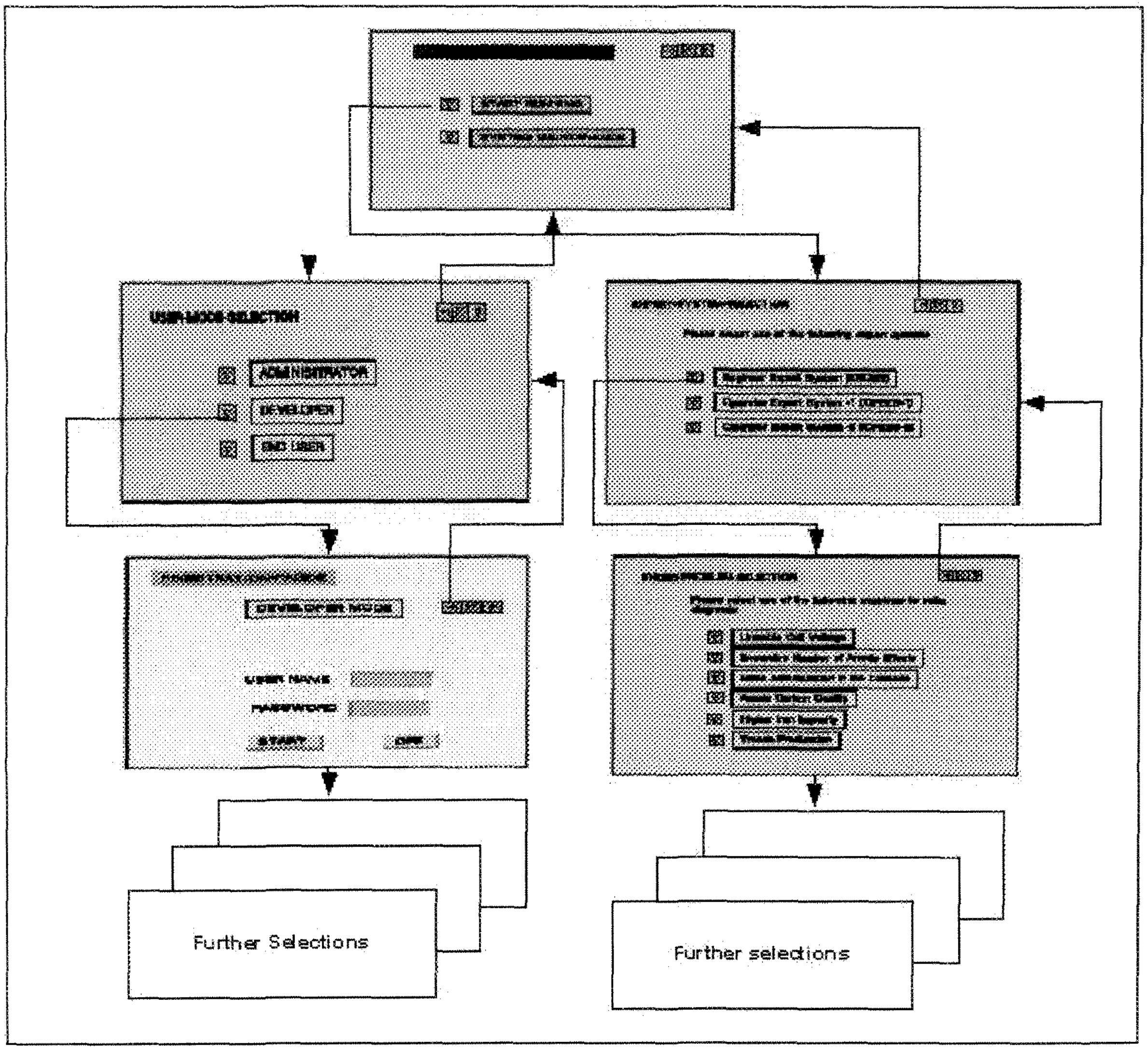

Figure 7.5 Example of system interfaces navigation

7.3.3.3 Example of display interface

As the system works in the on-line mode, the display of process data and status is a necessity. The user familiar spreadsheet style data file was used to edit and display the values and running status of relevant data, the spreadsheet is the standard function supported by GXL (G2 XL Spreadsheet). Figure 7.6 is an example, where the spreadsheet lists the process data read from data files. Although the features of GXL are similar to the conventional spreadsheet, some differences still were found from the testing results. The first is that data type is always fixed. Although many cell types can be presented on a single spreadsheet, the spreadsheet cannot randomly intersperse them. The second is about the definite size of spreadsheet; all the rows and columns must follow the specifications once the first one was established. After that the dimensions cannot be modified. These differences make some inconvenience for design work.

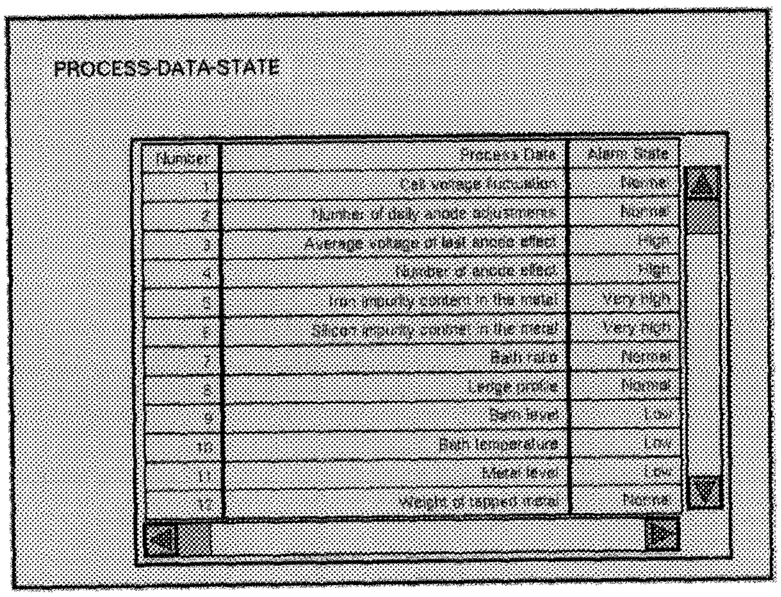

Figure 7.6 Example of process data alarm states

7.4 On-line diagnosis realization

One of the important features of G2, which was expected to improve the present system, is the real time communication capability, which is the weakness of Comdale/X. To realize the on-line diagnosis, the primary step is to choose the correct communication bridge, which does not only depend upon its performance but also depends upon the cost and the present situation of our laboratory. The second step is to use the selected bridge to access the data sources through the network, to test and to validate the designed system **[4, 5]**.

7.4.1 Communication bridge selection

According to the different cases of the real process, two sub-systems are designed to work in on-line mode. The corresponding reasoning processes are strongly depending upon the abilities of real time data acquirement. Thus, connectivity to external data sources is a central issue for these two sub-systems. G2 provides a wide range of communication bridges to gather the information from many sources, including database, plant floor equipment, etc.; that gives more possibilities of selection **[6, 7, 8, 9]**.

To choose a suitable bridge for the present system, two basic points must be discussed in advance:

- Based on the objective of the real time task of the present system, determine what kind of real time information is needed as the primary requirement.

- Analyze the features of available bridges to find out the most appropriate for the situation. The decisive factors of the selection must be concerned with both performance and cost.

As described in previous chapters, in the OPEES Comdale version, the system design does not attempt to directly connect with the plant floor instruments. All the process data, which were generated from the simulator or gathered from the databank of the real process, were stored in the data files, running as the data sources. In the G2 version, we do the same for the process data. Therefore, it is necessary to find a corresponding bridge to realize the communication between the G2 application and data files.

Based on the analysis of G2 communication tools, several relevant bridges are considered to meet the present system requirement. Consequently, to choose the most appropriate one, more detailed analysis about the techniques and reliability are needed. Three possible communication techniques are considered; they will determine what kind of data sources can be used. These techniques are based on the different external systems and the G2 communicating capacities:

- Enabling access to Intellution Fix Database, which is applied in the simulator system as the data source.
- Enabling communication between Windows based DDE application, which allows G2 to read and to write DDE data from Windows NT applications that support DDE or NetDDE.
- Enabling access to the COM (Component Object Model) application running under Microsoft Windows

Considering these techniques, the following communication tools of G2 are the potential choices:

- G2 Gateway
- G2 to Fix Bridge
- G2 to OPC Bridge
- G2 to DDE/NetDDE Bridge
- ActiveXLink

7.4.1.1 G2 Gateway

The G2 Gateway is a network-oriented toolkit used for developing the interfaces or bridges between G2 and other external systems. It allows the system to exchange various types of data between a G2 process and the bridge, which include databases, data acquisition systems, control systems, external simulation software, end-user displays and custom software applications. Figure 7.7 illustrates the possible connect functions; six types of bridges are provided for different data sources.

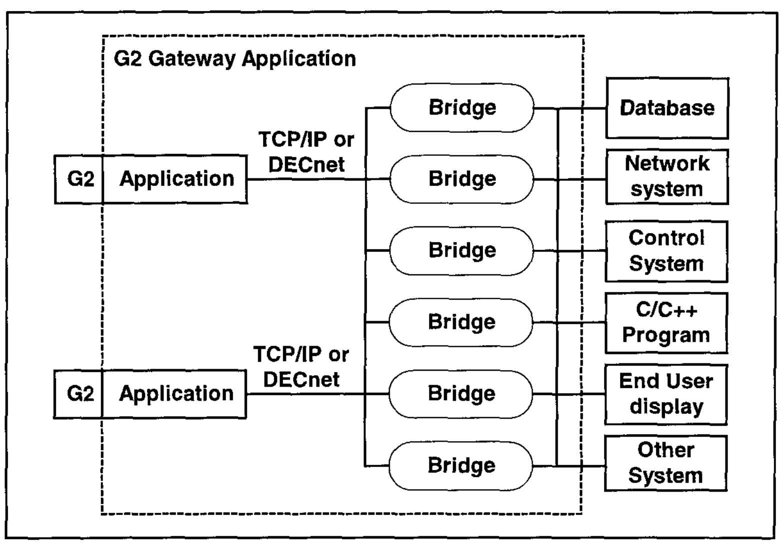

Figure 7.7 Communication types of G2 bridges

To access the database, the following bridges of G2 Gateway support the communication with the off-the-shelf databases:

- G2-Informix Bridge
- G2-Oracle Bridge
- G2-Rdb Bridge
- G2-Sybase Bridge
- G2-ODBC Bridge

Unfortunately, these standard bridges do not cover the three capacities mentioned at the end of the preceding section. That means the G2 gateway cannot be directly used to communicate with the desired data sources. Therefore another means must be considered to realize the communication with Intellution FIX database. The two possible communication bridges are discussed in the following.

7.4.1.2 G2-EDA Bridge

One of the available bridges to realize communication between G2 and FIX is the G2-EDA (Easy Database Access) Bridge, which enables the G2 application to access the process data from the Intellution FIX database. This bridge acts as an EDA client on behalf of G2, allowing the G2 application to simultaneously read and write data to the FIX. An independent software company, Matrikon Systems Inc., develops this bridge. Figure 7.8 shows the relationship between G2 and EDA.

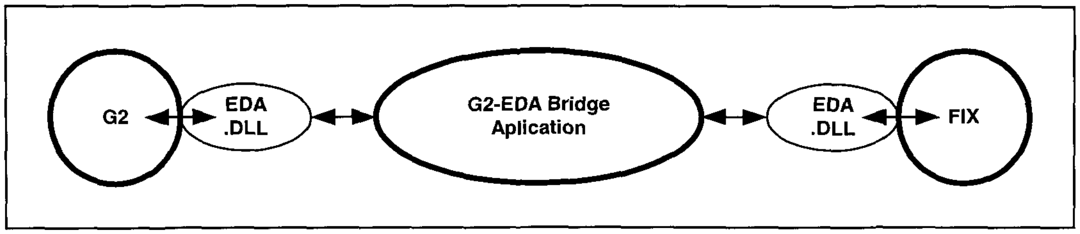

Figure 7.8 G2-EDA Bridge

7.4.1.3 G2-OPC Bridge

The other available bridge is the G2–OPC (OLE for Process Control) Bridge. This bridge enables the G2 application to access process data from any OPC compliant server, which is also developed by Matrikon Systems Inc. Figure 7.9 shows the system architecture of G2-OPC Bridge.

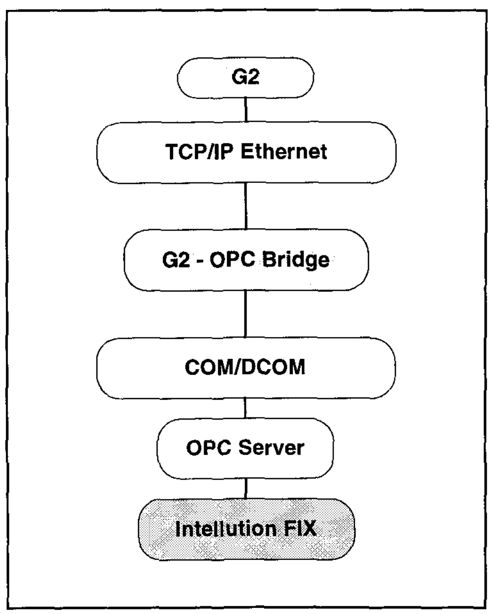

Figure 7.9 G2–OPC Bridge architecture

7.4.1.4 DDE/NetDDE Bridge

The third available bridge is G2-DDE/NetDDE that provides a communication between the Windows based application with the DDE server capability and G2. The DDE has been used to access the process data through the network in the Comdale version. Therefore, this will be used to shift the same type of the data files to the G2 version.

Although, all of these three bridges can meet the requirement, namely to allow the G2 application to access the process data by different ways, they were abandoned, not due to technical reasons but because their costs exceeded the budget. Therefore, attention shifted to the third possibility of communication bridges, which is G2 ActiveXLink.

7.4.1.5 G2 ActiveXLink

To connect the popular Microsoft Windows applications, G2 supports the common network communication tool: ActiveX technology, which is named G2 ActiveXLink. The G2 ActiveXLink is available for Windows NT and Windows 95/98 **[10, 11]**.

The ActiveX control is applied for the high-performance links to Microsoft Windows applications such as Microsoft Word, Excel, Visual Basic clients, Visual

C++, and Explorer Web browsers. The general connectivity of G2 ActiveXLink is illustrated in Figure 7.10.

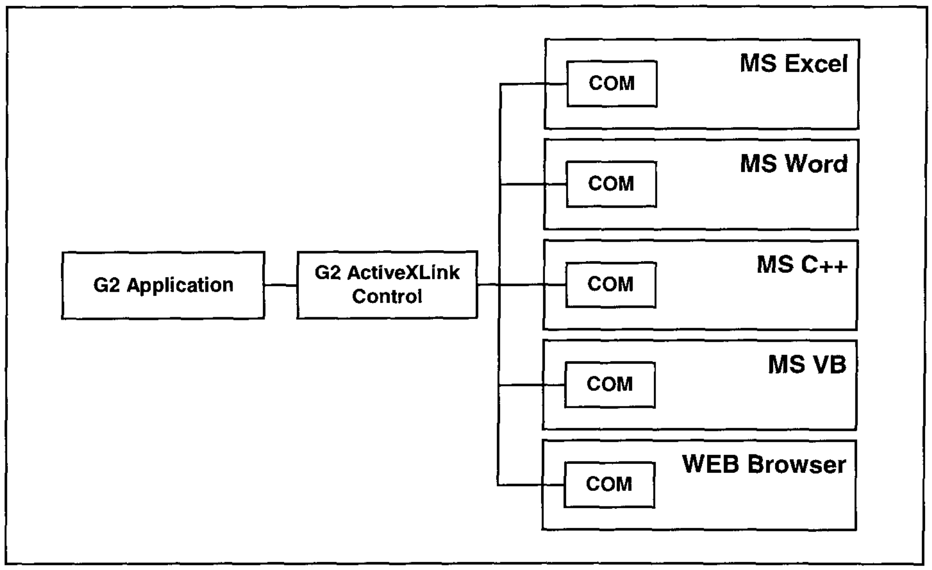

Figure 7.10 Connectivity of G2 ActiveXLink

As shown in Figure 7.10, the G2 ActiveXLink control performs as a bridge between the G2 application and COM-compliant applications. COM (Component Object Model) is a standard created by Microsoft and supported by Microsoft Office Products. Microsoft uses several names to describe this technology, including COM, DCOM (Distributed COM), and ActiveX. G2 ActiveXLink is based on the same standard Microsoft's ActiveX technologies, which allows a focus on the functionality of present applications rather than on the details of developing connectivity issues.

Finally, the G2 ActiveXLink was chosen to perform the bridge between the present G2 application and the data files in Microsoft Windows applications format. That decision was based on its features and the reasonable cost.

7.4.2 On-line diagnosis

To access the real time process data is the primary step to realize the on-line diagnosis of AEPES. Microsoft Excel is adopted as the data file to store and communicate the process data; this was also the case with the Comdale version.

G2 ActiveXLink allows the G2 to act as a server, so that the COM application can access it, and also allows G2 to access COM applications as a client. Therefore, the G2 ActiveXLink can be used to create a Microsoft Excel worksheet object, and also can be manipulated by using an interface that the Microsoft Excel application exports. Based on this characteristic, a procedure of the G2 application can be called or started by using the G2 ActiveXLink. The link between the G2 and Excel spreadsheet enables to be retrieved data from the G2 for display in a spreadsheet. Thus the present system can send its diagnosis results to an Excel spreadsheet on the user side to display the suggested message as consultation for the operators.

On-line data communication

To set up a communication bridge between the G2 application and the process data files, the following corresponding preset actions are considered:

- Add the control buttons to Excel spreadsheet and set the relevant properties.
- Then code a program in the G2 side to invoke the corresponding procedure.

After these preparations, the procedure in G2 can use the return values from the Excel spreadsheet.

Example of procedure with ActiveXLink

Here is an example of the procedure of the present system named:" *Get-process-data-value"*, which can execute the following functions:

- Invoke a procedure, which will call the values of process data from the Excel data spreadsheet.
- Send the return values to the relevant procedure of the present system, and allow the system to judge these values then release the corresponding alarm.

The example procedure is listed in Table 7.3:

Table 7.3 Example procedure

Get-process-data-value Procedure
Get-process-data-value () *G: class g2 com-interface;* *Ret: value;* *com-single: float* *begin* *if there exists a g2 com-interface G then begin* *Ret = Call g2 com-call ("Get-process-data-value") across G;* *Inform the operator that "Get return value of process data of cell operation: [Ret]";* *Ret = Call g2 com-call ("Get refresh data") across G;* *Inform the operator "The refreshed data have been received: [Ret]";* *end* *end*

7.4.3 Example of on-line diagnosis

By using the ActiveXLink control technology, the real time data and relevant messages can be more easily transferred from the expert system to the user side or in reverse. For example, the expert system and real process data can realize the communication through the network. Figure 7.11 shows an example of such information communication process:

- When OPEES-2 is ready to start the reasoning, the user can select the corresponding button to send a message to the process side to call for the required data.
- If the data file is in running status, it will return the required values to OPEES-2, and the inference engine will read this data and then do the reasoning.

- As the OPEES-2 is designed for the present plant situation: only a limited number of process data of cell are measured continuously. During the reasoning process, more information is needed. A check items list will be sent to the operator to ask for the required additional information, this information is mainly obtained by field checking or measuring.

- When the final problem has been found, the corresponding results will be sent back to the operator side to help them solve the problems. As this communication is in an interactive mode, the operator can also select relevant buttons to ask OPEES-2 to send back some messages, such as check items list or consulting result.

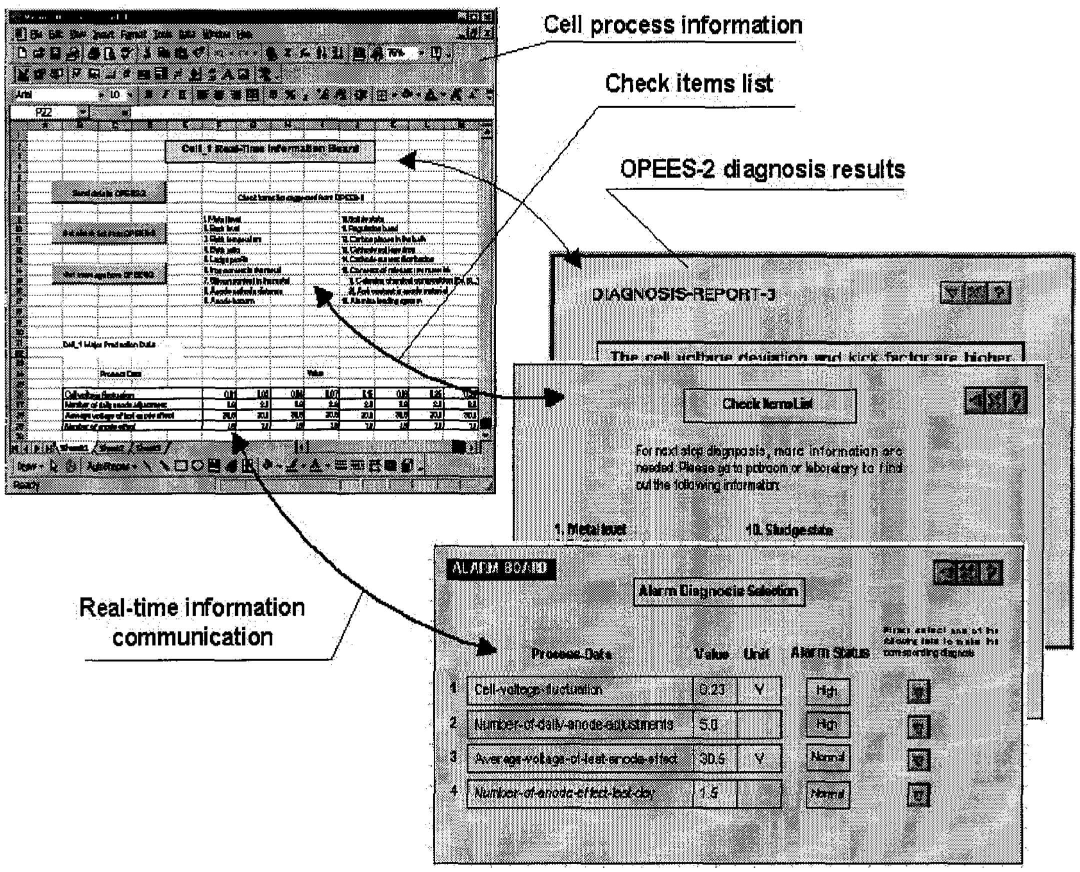

Figure 7.11 Communication between OPEES-2 and cell process

7.5 Results analysis

The emphasis of this design is to enhance the system performance with the G2's powerful development environment. Based on the primary testing results, the characteristics and explanations of this system can be summarized as follows:

System building

The present system development starts with only a few modules. After this basic system runs successfully, more modules are added to complete the system. Using this modularized approach yields a more reliable system structure that is easier to develop and to modify.

In principle, the G2 core and Utilities can provide the full range of the features to develop and deploy the present system, but for the particular functions, some additional G2 products are still needed, which were also suggested by the G2 Technical Support Department. For example, the GDA is recommended to handle the complex diagnostic procedures; it provides standard functions to monitor the changes in input values and informs the users when there are changes to the data they should know about. The real-time data treatment, display and relevant user interfaces are specially designed for the real applications. Without such additional G2 products support many useful functions would not be easy to implement and the connections between these functions would also be difficult to build.

Rule edition

Using the G2's natural language to develop the knowledge base, does not only mean we can use it to code the program like using human language, but also allows us to use the more effective rule statements to improve the reasoning process. For the event detection, as compared to the IF-THEN-ELSE rule, using the WHENEVER rule statement has been provided that can increase the reasoning speed.

Communication Bridge

In order to suit the various situations of the real cases, G2 supports wide possibilities of the communication tools to access the real time data. The selection process of the communication bridge of the present system shows that the G2's communication tools are flexible and powerful. This feature will help the system to better suit the different plant situations.

CHAPTER 8 CONCLUSIONS

The application of an expert system to the problem diagnosis and process control for the aluminum electrolysis process was studied. A two-level structure expert system was designed with a focus on the real process. The primary test results show that this expert system can be applied to the main types of cell to diagnose the most common problems and to assist operators in finding and solving such problems. The diagnosis results could also provide a reference to the control engineer to adjust the manipulated variables of the process control system. The results also show that the expert system offers possibilities to improve process productivity and quality, and can assume an important role in compensating for limitations of the conventional cell control system.

The knowledge applied for the knowledge base is acquired from the open sources and the domain experts. A major part of the knowledge is from experienced experts on electrolysis process problems diagnosis. Their experience provides the core of the present knowledge base, which can be used to diagnose the most common problems of the electrolysis process.

Considering that there are potential applications to different cases, to improve the flexibility of programming is an important part of system design. Several features of the present system could help to realize such goal. The modularized structure of the system makes it easier to program, modify and maintain in order to suit the needs of individual cells. The unique user interface helps to realize the interactive communication between the user and the system; it includes the structural interfaces and the functional interfaces, which in the text or graphic form; yield better comprehension and easier application.

In order to obtain correct reasoning results, the multiple symptoms criterion is adopted in the rule base programming.

This system can be selected to work in both off-line and on-line mode for the different cases. This is based on its unique structure, which consists of one off-line and two on-line sub-systems. When working in the on-line mode, the system can access the data file to read the real time data through the network. Depending on these real process data and with enough additional information, the expert system can do the reasoning and lead to the correct diagnosis result.

The test results show that this expert system has potentials in process supervision, process control and in operator training. The unique structure of the knowledge base and the design features allow this system to be readily applied and extended to different plants.

In summary, at this stage, the system realizes the on-line diagnosis, but only when connected with the virtual cell. As suggestion for further development, the system could be made to enhance its capacity of on-line diagnosis and parameter control and then could be connected to the real cell.

BIBLIOGRAPHY

Chapter 1

[1]. Tikasz, L., Bui, R. T., & Potocnik, V. (1990). Expert system applied to control of aluminum smelters. *TMS Light Metals*, 197-205.

[2]. Guid, G., & Tasso, C. (1994). *Design and development of knowledge-based system: from life cycle to development methodology*. Chichester, UK: John Wiley and Sons.

Chapter 2

[1]. Kammer, C. (1999). *Aluminum handbook*. Düsseldort, Germany: Aluminum–Verlag.

[2]. Horvath, J. (1998, February). Aluminum electrolysis. [Course material]. Budapest, Hungary.

[3]. Foosnæs, T., Grjotheim, K., Huglen, R., Kvande, H., Lilebuen, B., Mellerud, T., et al. (1986). *Understanding the Hall-Héroult process for production of aluminum*. Düsseldort, Germany: Aluminum-Verlag.

[4]. Tabereaux, A. (2000). Prebake cell technology: A global review. *JOM*, 52 (2), 22-28.

[5]. Haupin, W. (1995). Principles of aluminum electrolysis. *TMS Light Metals*, 195-203.

[6]. Grjotheim, K., & Welch, B. J. (1988). *Aluminum smelter technology – A pure and applied approach.* (2^{nd} ed.). Düsseldort, Germany: Aluminum–Verlag.

[7]. Arlyuk, B., (2000). Russia's aluminum industry after August 1998 crisis: status and prospects. *TMS Light Metals*, 129-135.

[8]. Feng, S., & Leng, Z. (2000). Development of high amperage prebake cells in China. *TMS Light Metals*, 171-178.

[9]. McLaughlin III, H., & Xing, G. (2000). Danjiangkou – A step forward. *TMS Light Metals*, 179-182.

[10]. Richards, N. E. (2000). Aspects of field studies on Hall-Héroult cells. *TMS Light Metals*, 195-201.

[11]. Kalban, A. J. M., Farsi, Y. A. M., & Binbrek, A. S. S. (2000). Reduction cell technology development at Dubal through 20 years. *TMS Light Metals*, 215-220.

[12]. Horvath, J. (1997, September). *Current trends in aluminum process metallurgy*. Paper presented at GRIPS (Groupe de Recherche en Ingénierie des Procédés et Systèmes) of Université du Québec à Chicoutimi. Chicoutimi, Québec.

Chapter 3

[1]. Durkin, J. (1994). *Expert systems design and development*. Englewood Cliffs, NJ: Prentice Hall.

[2]. Jackson, P. (1990). *Introduction to expert systems*. New York: Addison-Wesley Publishing Company.

[3]. Chris Naylor Research Ltd. (1999). XMaster – The easy expert system shell for windows. Retrieved August 21, 1999, from http://www.chrisnaylor.co.uk/AboutXMaster.html.

[4]. Parsaye, K., & Chignell, M. (1988). *Expert systems for experts*. New York, John Wiley & Sons.

[5]. Allen, J. (1994). *Nature language understanding*. Redwood, CA: Benjamin/Cummings Publishing.

[6]. Minsky, M. (1968). *Semantic information processing*. Cambridge MA: MIT Press.

[7]. Newell, A., & Simon, H. A. (1972). *Human problem solving*. Englewood Cliffs, NJ: Prentice Hall.

[8]. Davis, R. (1982). Expert systems: where are we? And where do we go from here? *AI Magazine*, 3(2).

[9]. Popovic, D., & Bhatkar, V. P. (1994). *Methods and tools for applied artificial intelligence*. New York: Marcel Dekker.

[10]. Luger, G. F., & Stubblefild, W. A. (1993). *Artificial intelligence, structures and strategies for complex problem solving.* (2^{nd} ed.). Redwood City, CA: Benjamin/Cummings Publishing.

[11]. Wenn, A. (2000). *Knowledge based systems.* [Subject Report]. School of Information Systems, Victoria University of Technology, Victoria, Austria.

[12]. Nii, H. P. (1993). *Tools and Infrastructure for Knowledge-Based Systems.* [Report]. International Technology Research Institute (ITRI) at Loyola College, Baltimore, MD.

[13]. Chou, J. Q. B. (1990). *Integration of expert systems and process simulation.* Paper presented at the workshop of *Expert system and simulation in plants*, Jonquiére, Québec.

[14]. Ignizio, J. P. (1991). *Introduction to expert systems - The development and implementation of rule-based expert systems.* New York: McGraw-Hill.

[15]. Muggleton, S. (1990). *Inductive acquisition of expert knowledge.* Reading, MA: Addison-Wesley Publishing.

[16]. Kashyap, N. (1997). *An expert systems application in electromagnetic compatibility.* Published Master thesis, University of Missouri-Rolla, Rolla, Mo.

[17]. Bartos, F. J. (1997, July). Artificial intelligence: smart thinking for complex control. *Control Engineering.* Retrieved October 3, 2001, from http://controleng.com/archives/1997/ctl0705.97/07f500.htm.

[18]. Owen, T. (1997, July). Artificial intelligence puts brain power in control. *Control Magazine.* Retrieved May 3, 2000, from http://www.controlmagazine.com/web_first/ct.nsf/articleid/mfey-4msm8t.

[19]. Guida, G., & Tasso, C. (Eds.) (1989). *Topics in expert system design, methodologies and tools.* Amsterdam, NL: North-Holland.

[20]. Ignizio, J. P. (1991). *An introduction to expert systems - the development and implementation of rule-based expert systems.* New York: McGraw-Hill.

[21]. Grjotheim, K. & Kvande, H. (1986). *Understanding the Hall-Héroult process for production of aluminum.* Düsseldort, Germany: Aluminum-Verlag.

[22]. DesBiens, C. (1992). *Base de connaissances pour la supervision de procédés.* Unpublished Master thesis. Université du Québec à Chicoutimi, Chicoutimi, Québec.

[23]. Harris, C. A., & Woo, E. (1988). Expert systems: application for process control. Paper presented at the *Canadian conference on computer applications in the mineral industry*, Québec City, Québec.

[24]. Jackson, P. (1990). *Introduction to expert system.* (2nd ed.). Wokingham, England: Addison-Wesley Publishing.

[25]. Kreutzer, W., & Mckenzie, B. (1989). *Programming for artificial intelligence, methods, tools and applications*. Boston, MA: Addison-Wesley Publishing.

[26]. Demeri, M. Y. (Ed.). (1990). *Expert system applications in materials processing and manufacturing*. Warren, PA: TMS.

[27]. Durkin, J. (1994). *Expert system design and development.* Englewood Cliffs, NJ: Prentice Hall.

[28]. Russell, S., & Norvig, P. (1995). *Artificial intelligence - A modern approach.* Englewood Cliffs, NJ: Prentice Hall.

[29]. Bender, E. A. (1996). *Mathematical methods in artificial intelligence.* Los Alamitos, CA: IEEE Computer Society Press.

[30]. Coyne, R. D., Rosenman, M. A., Radford, A. D., Balachandran, M., & Gero, J. S. (1989). *Knowledge-based design systems.* Reading, MA: Addison-Wesley Publishing.

[31]. Holtzman, S. (1989). *Intelligent design systems.* Reading, MA: Addison-Wesley Publishing.

[32]. Kreutzer, W., & McKenzie, B. (1990). *Programming for artificial intelligence, methods, tools and applications.* Singapore: Addison-Wesley Publishing.

[33]. Luger, G. F., & Stubblefield, W. A. (1989). *Artificial intelligence and the design for expert systems*, Redwood City, CA: Benjamin/Cummings Publishing.

[34]. Krijgsman, A. J., Verbruggen, H. B., & Bruijn, P. M. (1988). Knowledge-based real-time control. Paper presented at the workshop of *IFAC, Artificial intelligence in real-time control.* 13-19. Swansea, UK.

[35]. Årzén, K.E. (1988). An architecture for expert system-based feedback control. Paper presented at the workshop of *IFAC, Artificial intelligence in real-time control*, 21-26. Swansea, UK.

[36]. Handelman, D. A., & Stengel, R. F. (1988). Perspectives on the use of rule-based control. Paper presented at the workshop of *IFAC, Artificial intelligence in real-time control*, 27-32. Swansea, UK.

[37]. Øgå, O., & Woods, E. (1988). Intelligent alarm handling. Paper presented at the workshop of *IFAC, Artificial intelligence in real-time control*, 33-38. Swansea, UK.

[38]. Zinser, K. (1988). Design issues and knowledge representations for modern control room interface. Paper presented at the workshop of *IFAC, Artificial intelligence in real-time control*, 113-118. Swansea, UK.

[39]. Sanfeliu, A. (1989). Expert systems shells for automation. *Artificial Intelligence in Scientific Computation: Towards second-generation systems*. River Edge, NJ: Scientific Publishing Co., 43-50.

[40]. Dorn, J., & Doppler, C. (1996). Expert system in the steel industry. *IEEE Expert*, 18-21.

[41]. Bartos, F. J. (1997, July). Artificial intelligence: smart thinking for complex control. *Control Engineering.* Retrieved August 3, 2000, from http://controleng.com/archives/1997/ctl0705.97/07f500.htm.

[42]. Mahoudeaux, P. M. (1996, August). On-board and real-time expert control. *IEEE Expert*, 71-81.

[43]. Merritt, R. (1999, November). Artificial sentience? Embedded advanced controls, diagnostics, and optimization make today's control systems seem to think for themselves. *Control Magazine.* Retrieved August 3, 2001, from http://www.controlmagazine.com/ct/index.html.

[44]. Waterbury, B. (2000, November). Artificial intelligence expends frontiers in asset management. *Control Magazine.* Retrieved January 23, 2001, from http://www.controlmagazine.com/ct/index.html.

Chapter 4

[1]. Tikasz, L., Bui, R. T., & Potocnik, V. (1990). Expert system applied to control of aluminum smelters. *TMS Light Metals*, 197-202.

[2]. Li, J., Xiao, J., Liu, Y. X., Li, M. J., Shi, S. W., & Wang, F. (1997). Hierarchical intelligent control system for aluminum reduction cells. *TMS Light Metals*, 463-467.

[3]. Tikasz, L., Bui, R. T., & Horvath, J. (1995). Expert system-based consulting toolkit for aluminum electrolysis. Paper presented at the Conference of CIM (Canadian Institute of Mining, Metallurgy and Petroleum), Vancouver.

[4]. Rolland, W. K., Steinsnes, A., Larsen, A. S., & Paulsen, K. A. (1991). HALDRIS – An expert system for process control and supervision of aluminum smelters. *TMS Light Metals*, 437-443.

[5]. Zhang, J. P., Liu, L. P., Zhang, Y. F., Li, K. Y., Cheng, L. E., Gu, J. H, et al. (2000). New type control system for aluminum cell. *Metallurgical Industry Automation,* 24 (1), 12-14.

[6]. Consortium studies intelligent pot control. [Brief Report]. (2000, February). *JOM*, 4.

[7]. Grjotheim, K., & Welch, B. J. (1988). *Aluminum smelter technology.* Düsseldort, Germany: Aluminum-Verlag.

[8]. Larsson, J. E., & Persson, P. (1988). The knowledge database used in an expert system interface for IDPAC. Paper presented at the workshop of *IFAC, Artificial Intelligence in Real-Time Control*, 97-102. Swansea, UK.

Chapter 5

[1]. Haupin, W. (1999, September). Alumina properties. Paper presented at the short course of *TMS Industrial aluminum electrolysis – Theory & practice of primary aluminum production*, Seattle, WA.

[2]. Tabereaux, A. T. (1999, September). Bayer process. Paper presented at the short course of *TMS Industrial aluminum electrolysis – Theory & practice of primary aluminum production.* Seattle, WA.

[3]. Tabereaux, A.T. (1999, September). Cell design. Paper presented at the short course of *TMS Industrial Aluminum Electrolysis – Theory & practice of primary aluminum production.* Seattle, WA.

[4]. International Aluminum Institute. (2000, November). Aluminum production. *World Aluminum.* Retrieved October 3, 2001, from http://www.world-aluminium.org/iai/stats/650.html.

[5]. Welch, B. J. (1999). Aluminum production paths in the new millennium. *JOM,* 51(5), 24-28.

[6]. Hale, W.R. (1990, April). Cell control and feed technology. Paper presented at the short course of *TMS Industrial aluminum electrolysis – Theory & practice of primary aluminum production.*, Pittsburgh, PA.

[7]. Desclaux, P. (1987). AlF_3 Additions based on bath temperature measurements. *TMS Light Metals*, 309-313.

[8]. Homsi, P., Peyneau, J. & Reverdy, M. (2000). Overview process control in reduction cells and potlines. *TMS Light Metals*, 223-230.

[9]. Tabereaux, A. T. (1992, April). Diagnosis and correction of irregular cells. Paper presented at the short course of *TMS Industrial aluminum electrolysis – theory and practice of primary aluminum production*. Pittsburgh, PA.

[10]. Horvath, J. (1999, August). *Report on visit at LECAP (Laboratoire expérimental de contrôle avancé des procédés)*. [Report]. Université du Québec à Chicoutimi. Chicoutimi, Québec.

[11]. DesBiens, C. (1992). Base de connaissances pour la supervision de procédés. Unpublished Master thesis. Université du Québec à Chicoutimi.

[12]. ALCAN Inc., (1991). "EPURANXP" & "VS-ANODE" Expert system. [Testing Program].

[13]. Tabereaux, A. T. (1997, September). Diagnosis and correction of irregularly operating cells. Paper presented at the short course of *TMS Industrial aluminum electrolysis, Theory & practice of primary aluminum production.* Charleston, SC.

[14]. Tabereaux, A. T. (1999, September). Cell operations and diagnostics. Paper presented at the short course of *TMS Industrial aluminum electrolysis, Theory & practice of primary aluminum production.* Seattle, WS.

[15]. Tabereaux, A. T. (2001, September). Fundamentals of cell operations and diagnostics of irregularly operation cells. Paper presented at the short course of *TMS Industrial aluminum electrolysis, Theory & practice of primary aluminum production.* Louisville, KY.

[16]. Neuron Data, Inc. (1991). *NEXPERT OBJECT User's guide*. Palo Alto, CA.

[17]. Neuron Data, Inc. (1991). *NEXPERT OBJECT Manual.* Palo Alto, CA.

[18]. Mann, V. K., Yurkov, V. V., Polyakov, P. V., & Buzunov, V. Y. (1998). Cryolite and bath temperature stabilization problem in aluminum reduction cell. *TMS Light Metals*, 371-377.

[19]. Enter, P.M. (1998). ELYSESEM - An education and information tool. *TMS Light Metals*, 341-348.

[20]. Caissy, J., Dufour, G., & Lapointe, P. (1998). On the road to 325 kA. *TMS Light Metals*, 15-219.

[21]. Sturm, E., & Wede, G. (1998). Removing impurities from the aluminum electrolysis process. *TMS Light Metals*, 235-240.

[22]. Madsen, D.J. (1991). Temperature measurement and control in reduction cells. *TMS Light Metals*, 453-456.

[23]. Aiquel, G. & Craig, R. (1991). Controlled operational transitions in P-195 cells. *TMS Light Metals*, 379-382.

[24]. Cypert, D. E., Forrester, F. M., Hu, A. C., McCafferty, C. T, Parmer, R. E., Taylor, J. B. et al. (1992). Implementation of a distributed control system for reynolds P-195 potline at ALCASA smelter, Venzuela. *TMS Light Metals*, 347-353.

[25]. Campo, J. J. D., & Blanco, F. (1995). Decision making algorithm for potroom operation. *TMS Light Metals*, 413-421.

[26]. Levening, B. (1995). ELAS-VAW's cell-control automation system. *TMS Light Metals*, 387-389.

[27]. Tang, H. Q., Uratd, N., Read, C. M., & Stejer, S. L. (1998). Response of a Hall-Héroult cell to step - Changes in operating conditions: measurements and "dynamic" simulations. *TMS Light Metals*, 349-357.

[28]. Fryer, G. G., Nichol, R. G., & Rieg, J. N. (1998). Reduction cell control technology. *TMS Light Metals*, 331-339.

[29]. Vallés, A., Aiquel, G., & Craig, R. (1989). Anode setting height optimisation. *TMS Light Metals*, 479-482.

[30]. Salt, D. J. (1990). Bath chemistry control system. *TMS Light Metals*, 299-304.

[31]. Entner, P. M. (1995). Control of bath temperature. *TMS Light Metals*, 227-230.

[32]. Gadd, M. D., Welch, B. J., & Ackland, A. D. (2000). The effect of process operations on smelter cell top heat losses. *TMS Light Metals*, 231-237.

[33]. Kvande, H. (1999). *Cell voltage, anode effect*. Paper presented at the conference of *18th International Course on Process Metallurgy of Aluminum*, (pp.10.1-10.10). Trondheim, Norway.

[34]. Thonstad, J., Utigard, T. A., & Vogt, H. (2000). On the Anode Effect in aluminum electrolysis. *TMS Light Metals*, 249-256.

[35]. Li, X. B., Yuan, Z. T., & Yang, Z. (1999). Developemnt of an automatic control systsem of alumina dense phase and hyper dense phase storage and covey. *Journal of Guansu University of Technology*, (1).

[36]. Girard, S., Simard, M., & Tremblay, J. P. L. (1987). Computer control system for the A-275 Cell. *TMS Light Metals*, 189-197.

[37]. Huni, J. P. R. (1987). A-275 – Individual anode control. *TMS Light Metals*, 199-202.

[38]. LoBmann, G. (1992). Utilization of various measurements in reduction cells for operational improvement. *TMS Light Metals*, 441-447.

[39]. León, D. D., & Hu, H. (1991). Application of statistical process control technique in process monitor development. *TMS Light Metals*, 903-908.

Chapter 6

[1]. Tabereaux, A. T. (1992, April). Diagnosis and correction of irregular cells. Paper presented at the short course of *Aluminum electrolysis – Theory &practice of primary aluminum production.* Pittsburgh, PA.

[2]. Tabereaux, A. T. (1997, September). Diagnosis and correction of irregularly operating cells. Paper presented at the short course of *TMS Industrial aluminum electrolysis, Theory & practice of primary aluminum production.* Charleston, SC.

[3]. Tabereaux, A. T. (1999, September). Fundamentals of cell operations and diagnostics of irregularly operation cells. Paper presented at the short course of *TMS Industrial aluminum electrolysis, Theory & practice of primary aluminum production.* Seattle, WS.

[4]. Tabereaux, A. T. (2000, May). Problèmes d'opération des cuves. Paper presented at the symposium of *Symposium sur le procédé d'électrolyse 2000, (3e éd),* Bécancour, Québec.

[5]. Horvath, J. (1999, August). *Report on visit at LECAP (Laboratoire expérimental de contrôle avancé des procédés).* [Report]. Université du Québec à Chicoutimi, Chicoutimi, Québec.

[6]. Tikasz, L., Bui, R.T., & Potocnik, V. (1994). Aluminum electrolytic cells: A computer simulator for training and supervision. *Engineering with computers*, No.10. 12-21.

[7]. Tikasz, L., Bui, R.T., & Villeneuve, V., & Doyon, S. (2001). From cell models to virtual potroom: A tool for operation and research. *TMS Light Metals*, 1251-1257.

[8]. Meghlaoui, A., Mohammed, Y. A., & Tikasz, L. (1998). Adaptation of an electrolytic cell model in static and dynamic modes, *TMS Light Metals*, 403-408.

[9]. Durkin, J. (1994). *Expert systems, design and development*. Englewood Cliffs, NJ: Prentice Hall.

[10]. Voorbraak, F. (2000). Uncertainty and vagueness, knowledge-based systems, Lecture 5. University of Amsterdam. Amsterdam, Netherlands.

[11]. Parsaye, K., & Chignell, M. (1988). *Expert systems for experts*. New York, NY: John Wiley &Sons.

Chapter 7

[1]. Gensym Corporation. (1997). G2 Reference manual. (Vols.1, 2). (Version 5.0). Cambridge, MA.

[2]. Gensym Corporation. (1997). G2 GUIDE - User's guide, G2 utilities. (Version 5.0). Cambridge, MA.

[3]. Gensym Corporation. (1997). G2 Spreadsheet - User's guide, G2 Utilities. (Version 5.0). Cambridge, MA.

[4]. Gensym Corporation. (1997). G2 Gateway - Bridge developer's guise. (Version 5.0). Cambridge, MA.

[5]. Gensym Corporation. (1997). Telewindows - User's guide. (Version 5.0). Cambridge, MA.

[6]. Mintchell, G. A. (2001, January). OPC integrates the factory floor. *Control Engineering.* Retrieved August 3, 2001, from http://controleng.com/archives/2001/ctl0101.01/010101.htm.

[7]. Laurentide Controls, Inc. (2000, February). OPC performance testing. [White Paper]. Montréal, Québec.

[8]. Matrikon System Inc. (1999). Gensym G2 bridge for Intellution fix. Edmonton, Alberta. Retrieved September 21, 1999, from http://www.matrikon.com/downloads/systems/manuals/g2eda.pdf.

[9]. Matrikon System Inc. (1999). Gensym G2 – DDE/NetDDE bridge. Edmonton, Alberta. Retrieved September 21, 1999, from http://www.matrikon.com/downloads/systems/manuals/g2dde.pdf.

[10]. Gensym Corporation. (1998). G2 ActiveXlink - User's guide. (Version 1.0.). Cambridge, MA.

[11]. Tanenbaum, A. S. (1996). *Computer networks.* (3rd ed.). Upper Saddle River, NJ: Prentice Hall.

[12]. Baltar, M. S., Silva, E. F. V., & Filho, C. S. (1998, August). Analyzing and monitoring aluminum smelter performance via web browsers. *JOM*, 14 -16.

[13]. Drakulick, M. (1999, September). Data acquisition software – Maintaining your plant's health. *Control Engineering.* Retrieved August 3, 2001, from http://controleng.com/archives/1999/ctl0901.99/990910.htm.

[14]. Mintchell, G. A. (2000, July). Web technologies power automation interface software. *Control Engineering.* Retrieved September 24, 2001, from http://controleng.com/archives/2000/ctl1202.00/e0012p23.htm.

[15]. Montague, J. (2000, October). How to build an industrial network. *Control Engineering*. Retrieved December 15, 2001, from http://controleng.com/archives/2000/ctl1001.00/001001.htm.

[16]. Montague, J. (2001, January). Simplify your network with better tools, common sense. Retrieved July 23, 2001, from http://controleng.com/archives/2001/ctl0101.01/010104.htm.

[17]. Stanley, G.M., & Vaidhyanathan, R. (1998, June). A generic fault propagation modeling approach to on-line diagnosis and event correlation. Paper presented at *the 3rd IFAC Workshop on on-line fault detection and supervision in chemical process industries*, Solaize, France.

[18]. Chang, D. H. (1999). On-site real-time data acquisition for real-time database of fault diagnostic expert system. *Process Automation Instrumentation*, 20 (2).

[19]. Alford, L., Cairney, C., Hoggs, R., Honsowetz, M., Huynh, V., Jines, A., et al. (1999, July). Online expert-system applications use in fermentation plants. *Expert Systems.* Retrieved December 12, 1999, from http://www.isa.org/journals/intech/feature/1.1773.199.00.html.

Appendix.1

[1]. Comdale Technologies (Canada), Inc. (1995). Comdale/X - User's manual and reference guide. (Version 5.12). Toronto, Ontario.

[2]. Comdale Technologies (Canada), Inc. (1998). SmartWorX - User's guide. (Version 1.03). Toronto, Ontario.

[3]. Gensym Corporation. (1997). G2 Developer's guide. (Version 5.0). Cambridge, MA.

[4]. Comdale Technologies (Canada) Inc. (1998, September). Real - time expert system control system for Cerro Matoso, S.A. [White Paper]. Toronto, Ontario.

[5]. Gensym Corporation. (1999). "Alcoa filters out problems in alumina processing unit using G2". [White Paper]. Retrieved January 23, 2000, from http://www.gensym.com/successstories/alcoa.html.

Appendix.2

[1]. Neuron Data, Inc. (1991). NEXPERT OBJECT User's guide. Palo Alto, CA.

[2]. Neuron Data, Inc. (1991). NEXPERT OBJECT Manual. Palo Alto, CA.

[3]. Arvida Research and Development Center. Alcan International Ltd. (1992). "EPURANXP" & "VS-ANODE" expert system. [Testing Program]. Jonquiére, Québec.

Appendix.3

[1]. Comdale Technologies (Canada) Inc. (1995). Comdale/X - User's manual and reference guide. (Version 5.12). Toronto, Ontario.

[2]. Comdale Technologies (Canada) Inc. (1998). SmartWorX - User's guide. (Version 1.03). Toronto, Ontario.

APPENDIX

APPENDIX 1: APPLIED EXPERT SYSTEM SHELLS STUDY

A-1.1 Comdale/X

A-1.1.1 Introduction to Comdale/X

Comdale Technology Inc. is a Canadian company that has provided AI software for industrial process control since 1986. It developed the tools to monitor and control both discrete and continuous manufacturing processes. Their suite of products includes Comdale/X, Comdale/C, and ProcessVision. In 1998, Comdale released its second-generation product: SmartWorX Suit. At the end of 1998, Comdale Technology was acquired by ABB Ltd., which is committed to provide support for products of Comdale.

The applied version for the present expert system is Comdale/X V5.13. Although several versions have been developed, the basic structure and programming approach are identical. The main characteristics are as follows **[1]**:

- Comdale/X is an off-line consultative expert system, which requires the input from the user for information required in making its decisions. Comdale/X is included with Comdale/C as the development tool for real-time expert systems. Comdale/X has the capability to incorporate hypertext documents with the reasoning abilities of the expert system to produce expert hyper manuals, which provide information and generate advice through the interface.

- Comdale/X is a rule-based shell that provides both forward and backward chaining. The inference control strategies manage the search, focus, and conflict resolution mechanisms. The search mechanism allows for the rule search to be done either depth-first or breadth-first. The focus mechanism provides various options for finding the optimum solution path, including highest certainty, lowest certainty, highest priority, or lowest priority. When more than one rule may apply, mechanisms, including alphabetic mechanisms, and various priorities may be used to solve this conflict.
- Expert system knowledge in the knowledge base is represented as objects, classes, rules, or procedures:
    - Objects are expressed as object-attribute-value triplets and can be of different types, including logical, string numerical, date, and time.
    - Facets may be added to the triplets to specify information, such as uncertainty, time, fuzzy sets, customized questions, and sources of information.
    - Classes group similar objects into a hierarchy. Comdale/X classes have full inheritance capabilities, including multiple inheritance and both public and private attributes.
    - Rules are expressed as if-then-else statement and are used to encode the concepts, mathematical expressions, time, and string expressions.
    - Certainty factors may be attached to rules to reflect the confidence in the conclusion of the rule.
    - Procedures determine the control of rule execution and class/object manipulation.
- Comdale/X also provides some facilities for the construction of the knowledge base and debugging of the system:
    - Graphical editors are used for the classes, objects, rules, procedures, rule-sets, and mapping-files editing.
    - Class/object browser shows the hierarchical relationships between classes and objects and allows for their direct modification.
    - Rule browser shows the interrelationships among rules and allows rule editing to be done.
    - Debug includes cross-referencing triplets, tracing rules, and watch variables.

Comdale/X can be stand-alone or be an integrated part of other applications. To combine with others to address a specific solution, Comdale provides bridges to connect the application software, data souses, I/O devices and others. Figure A-1. 1 shows the available applications of Comdale.

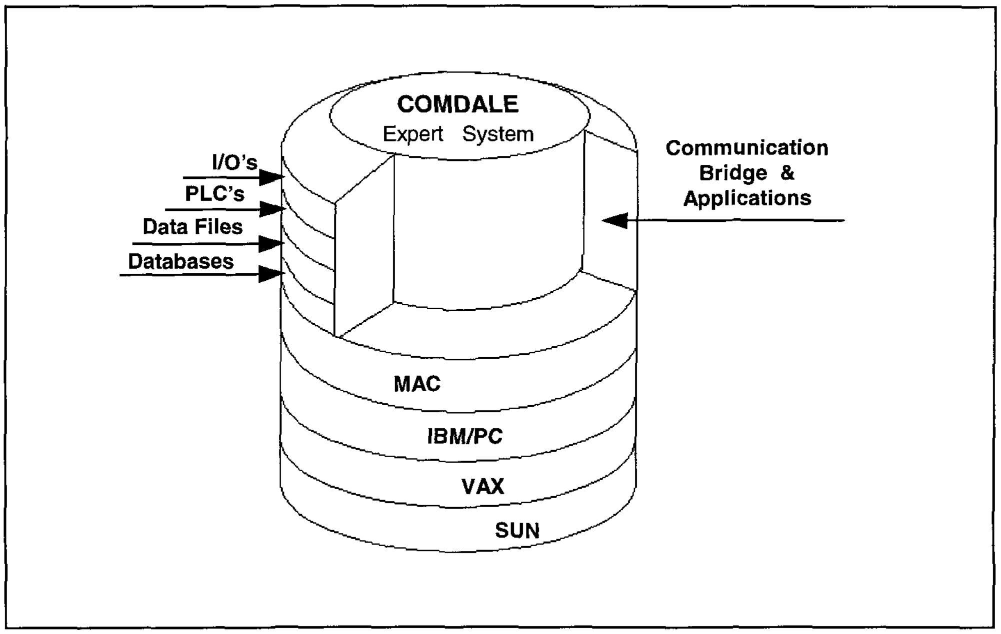

Figure A-1. 1 Comdale expert system applications structure

In summary, an expert system created by using Comdale/X will have the following abilities:

- Capture knowledge - which will make it possible to retain the knowledge from the domain experts.
- Evaluate knowledge - after expert system has been done, it allows the experts to test and verify the acquired knowledge.
- Diagnose problems - depending on the information given, arrive at conclusions and give advice.

- Obtain the relevant information manually - it can ask questions and identify the concerned area and present the important information by the interface.
- Training tool - it will quickly transfer the knowledge needed to perform certain tasks from the knowledge base to an inexperienced person. It allows the user to learn and understanding the subject matter.

Comdale/X consists of two programs, namely the Comdale/X Rule Compiler and the Comdale/X Application Program. All knowledge in the Comdale/X Knowledge Base is in the form of IF-THEN rules. The Comdale/X Rule Compiler is used to compile these rules and to generate reports/output files. It allows the knowledge engineer to represent, test, debug and enhance the knowledge contained in the expert system.

The Comdale/X graphical user interface allows the operator to use the knowledge for decision-making. It also provides a means for the facility to request explanations regarding conclusions the expert system has made. All conclusion statements in each rule have certainty factors associated with them to emulate the fuzzy thinking of humans.

A-1.1.2 New generation of Comdale

The SmartWorX is the new generation of Comdale/X, which was developed in 1998. Once it was selected as a strong candidate for the further development of the present system. The SmartWorX has been tested for verification and comparison. Some preparations for software transformation have been done.

The SmartWorX is a Windows NT-based Suite. The brief features of SmartWorX are enhanced real-time performance, intelligent real-time alarm detection, neural networks, and optimized knowledge development to improve the previous Comdale/X **[2]**.

The SmartWorX Suite includes three parts:

- <u>SmartWorX Alarm Manager</u> is an intelligent real-time alarm detection and management system.
- <u>SmartWorX Knowledge Developer</u> is a real-time expert system development tool.
- <u>SmartWorX Expert Optimizer</u> standardizes operating practices to constantly target and achieve process stability and optimization.

Although the SmartWorX was not finally applied for the present system, its basic functions were studied and some testing programs have been coded, which benefited the further system development in the following aspects:

- How to design a more utility alarm form, which can automatically scan the available process variables and lists detected status.
- Design a user-modifiable function, which can be taken on alarm limitation. This assists the operators with individually situation of cells.
- Use graphical user interface to assist operators to understand the final diagnosed result of why the abnormal case occurred.
- Due to the extensive adoption of Microsoft standards, using the relevant applications as the data files.
- The real-time development tool describes the basic communication technologies, which helps the further communication bridge selection.

A-1.2. Gensym G2

A-1.2.1 Introduction to Gensym G2

Gensym is a major company that provides the software products and services of the expert system. The G2 is Gensym's flagship product; it is a powerful development and deployment environment for managing and optimizing dynamic, complex decision support and control applications. The G2 offers a graphical, object-oriented environment for creating intelligent applications that monitor, diagnose, and control dynamic events in on-line and simulated environments. It allows the user to express objects, rules, methods and procedures by using the structured natural language. So, it can be used for rapidly creating an application that reasons with real-time operations data. The G2 applications can follow multiple lines of reasoning and analyze large amounts of data and numerous trends concurrently. Therefore, it can be used for a real-time system and for connections with the wide selection of databases, PLCs, DCSs, real-time data systems, standard MS Office applications and other systems through ActiveX, CORBA, Java, C/C++ and etc.

Generally, G2 improves the performance of an operation by the following approaches:

- Continuously monitors the potential problems and takes the alarm before they adversely affect the operation.
- Takes the complex operations data as one segment of the useful information for the reasoning by using knowledge-based models and analysis.

- Diagnosing the root cause of time concerned problems and taking the corrective actions.
- Maintain the optimal operating conditions.
- Coordinates the activities and information in complex operation processes.
- Communicates through the network management between G2 applications to an external system.
- Offers the enhanced ability for process design, simulation, supervision and advanced control.

A-1.2.2 General structure of G2

The G2 environment consists of four major areas:

- G2 Application Server
- Telewindows client
- G2 Utilities
- G2 Gateway Bridge

The G2 Application Server also called the G2 core, provides the full range of features needed to develop an intelligent real-time applications. The diagram of the G2 application server is shown in the Figure A-1. 2 **[3]**.

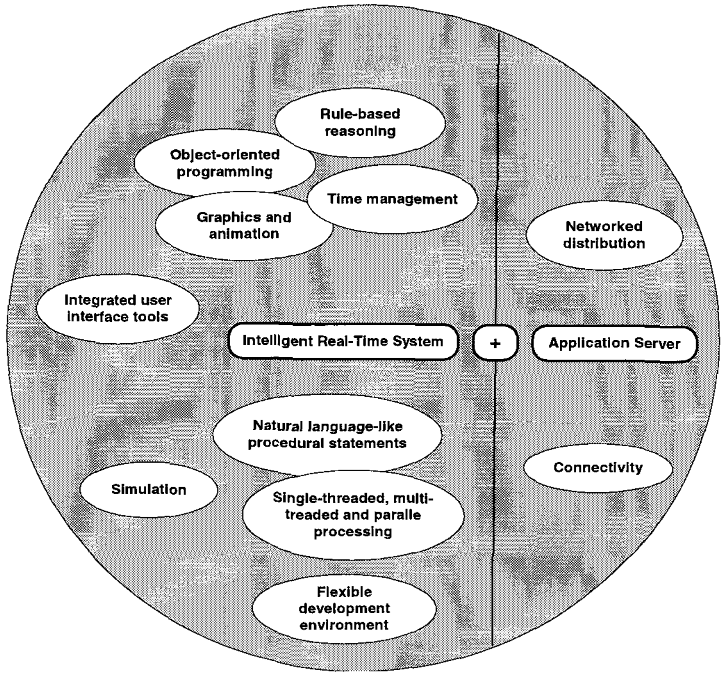

Figure A-1. 2 G2 application server

G2 Telewindows client allows several users on the network to access the server application from other machines, where the client can view or manipulate the application such as using the server's terminal. This environment is facilitated for the knowledge engineer or supervisor to develop, maintain or monitor the system from different places.

G2 Utilities are the optional components of the G2 application server that allow user to perform specific functions. They are the necessary tools to create a customizable knowledge base, interface, communication handling, and documentation.

G2 Gateway Bridge integrates many different software technologies together in one package, which are concerned with: object technology, knowledge-base technology, data interfaces, graphical user interface and application server support. Through the G2 Gateway Bridge, a wide variety of data sources, such as database, control systems and various real-time data sources can be reached.

Except for the G2 Application Server, which is the main tool used for the present system development, the other application products are also used in the system design. For example, the GUIDE of G2 Utilities was used to design the graphical interfaces, such as the user selection menu, question window, and alarm status. The ActiveXLink Communication Bridge is used to establish communications between G2 and external data souses.

To perform the complete functions of G2, several layered products are also needed for development in the specific domains. For example, to design the present system, the GDA (G2 Diagnostic Assistant) is a very useful tool that enables data monitoring, filtering, diagnosing, statistical process control, alarm management, combinatorial and fuzzy logic, close-loop control. The principle component of GDA is a graphical language that allows expressing complex diagnostic procedures as a diagram of blocks. These diagrams can acquire data from real-time process, then make inferences based on this data. Depending upon the inference results, it can take actions such as raising alarms, sending messages to operators, or concluding new set points. Unfortunately, this product was not covered by the G2 license we purchased; that led to a difficult situation to realize such functions by using the general elements functions of G2 core.

A-1.2.3 Using new design techniques to develop system

To design the knowledge base, G2 provides some new techniques, which differ with the traditional design techniques. The traditional design techniques concerned with five stages: analysis, design, implement, testing and deployment. Generally, these stages are processed sequentially, however that caused some difficulties to system development. For instance, the end user requirements in the last stage seldom have an opportunity to provide feedback on the implementation until it is complete.

Considering these difficulties, G2 supports new style design techniques, and it can increase the flexibility of the application development. The developer can create a prototype that supports the basic needs of the application and then refines the prototype given input from the knowledge engineers and users.

This approach is based on the G2 development techniques standards, such as the modularized organization, object-oriented design and standardized user interface. The benefits of using such standards will increase the flexibility, maintainability and expendability of the system development. The detailed description about how to develop the present system by using such techniques is in Chapter 7.

A-1.3 Application examples study

Two expert system shells, Comdale/X and Gensym G2, are applied for the present system development; to study the successful examples of their applications is a direct way to obtain the experiences, especially for industrial application. Although these shells have been widely applied in industrial area, few of the application examples were found in aluminum industry. In these examples **[4]**, **[5]**, the interested points concluded are as follows:

- How to design an expert system with optimized target value
- How to evaluate and document the control procedures and operators' heuristic
- How to put the expert system into the real process
- How to use the G2 communication bridge to connect the existing DCS.

APPENDIX 2_VS-ANODE SYSTEM ANALYSIS

A-2.1 Introduction to NEXPERT shell

The NEXPERT is an expert system development tool with graphical user interface that was developed by Neuron Data Inc. The NEXPERT development system features a rule based and object based inference engine and provides a unique environment. It allows for the interfacing with databases, programming languages and other applications [1, 2].

A-2.2 VS-ANODE test result analysis

Three steps were executed to test the VS-ANODE expert system. First, learn the operation of the NEXPERT shell and run the VS-ANODE for testing purpose [3]. Second, analyze the knowledge base to search for the useful knowledge for the present expert system. Finally, compare with the Comdale/X syntax to find a way to transfer the selected rules in Comdale/X format.

A-2.2.1 Syntax comparison

The syntax of the rules of NEXPERT is written as **If...Then... and Do...** format. Where the **If** is followed by a set of conditions, the **Then** by a hypothesis or goal which becomes true when the conditions are met, and the **Do** by a set of actions to be undertaken as a result of a positive evaluation of the rule (conditions). But in Comdale/X shell, the rules are written in an **IF-THEN-ELSE** format. The **IF-AND-OR** part of the rule is called the premise. The **THEN-ELSE** part is called the conclusion. It is obvious that the syntaxes of the rule formats of these two shells are similar which will help us to comprehend the NEXPERT rules and make it easier to convert them into Comdale/X format.

Table A-2. 1 is an example taken from the rule base of NEXPERT, and Table A-2. 2 is the corresponding rule, which is converted into the Comdale/X format.

Table A-2. 1 Example rule of NEXPERT

@ RULE = R34
(@ LHS = *(Is (control.operation) ("Top Surface of the anode"))* *(Is (Ifm_raw_materials_usedl.status) (FALSE, NOTKNOWN))* *(Is (Ifm_manufactured_materialsl.status) (FALSE, NOTKNOWN)))* *(@ HYPO = control.nexpalnantion)* *(@ RHS =* *(Execute ("Write To") (@STRING=@TEXT=Sorry, but not explanation could be found for your\observation regarding the anode top surface.";)))*

Table A-2. 2 Example rule converted in Comdale/X format

```
@ Name = R34

IF control_operation.taxt.@string is "Top Surface of the Anode"
        AND fm_raw_material_used.status.false I
              fm_raw_material_used.status.unknown is TRUE

AND fm-manufactured_material.status.false I
      Ifm-manufactured_material.status.notknown is TRUE

THEN FORM ("HYPO_control_explanation.frm")
THEN TEXT ("Sorry, but not explanation could be found for your \
              observation regarding the anode top surface.")
```

A-2.2.2 Conclusion

Based on the analysis of the test results, several conclusions are obtained:

- The VS-ANODE expert system was designed only for the particular Söderberg anode and the specific parameters are required.
- Most questions request the numeral values of the relevant parameters. Without these necessary values, the reasoning process will discontinue.
- For special cases, VS-ANODE uses some text questions instead of the requirement of numerical parameters. But using such text questions cannot obtain the final reasoning result, only some suspected results are provided.
- Only partial rules are concerned with the text questions, which could be used as a reference for the present expert system.

APPENDIX 3 REASONING PROCESS IMPROVEMENT

During the development of AEPES, several approaches were adopted to improve the efficiency of diagnosis and the correctness of the reasoning result. The main steps of this development process were concerned with the following aspects:

- Knowledge base structure improvement
- Certainty factor utilization
- Improve diagnosis speed
- Multiple symptoms criterion
- Interface improvement.

The examples of the particular case studies about these aspects are presented in the following sections.

A-3.1 Knowledge base structure improvement

As described in **Section 6.4.3 "Modularization of knowledge base,"** two types of module are used for the present knowledge base building. The major purposes of using modules to organize the rule base is to separate the general knowledge and the specific knowledge, as well as to improve the knowledge base programming work. No doubt, with such modulized knowledge base, the maintenance work will be simplified.

For example, in the total 298 rules of the VSS sub-system of OPEES-1, there are 83 rules concerned with the on-line process data communication, data status detection and alarm release. All these rules belong to the alarm module, which is a unique module concerned with the specific knowledge of the particular cells. The general knowledge is not varied for the particular cell applications; they are organized into other types of modules, which are not concerned with the particular value of process data.

Based on such structure of the knowledge base, the only modification work for the extension application is concerned with the part of the alarm modules. The whole structure of the knowledge base and the general knowledge modules will be retained. The routine updating, which concerns the variation of specification of the particular cell, is only needed to modify the limitations of the alarm modules through the system provided special interface. The following examples will describe the working procedure of the alarm module, which is the only part of the knowledge base concerned with the specific knowledge.

A-3.1.1 Example rule of on-line communication setting

Here is an example of a rule of the alarm module of VSS sub-system, which is concerned to setting the connection format of the data files (See Table A-3. 1). This is the primary step to communicate with the real process.

Table A-3. 1 Rule alarm1

Rule
@name = alarm1
@priority = 5

```
IF   cell.selected.@string
THEN trl.TYPE.@string is "DDE"
THEN trl.MAPFILE.@string is "D:\comdale\exp_2.mpf"
THEN trl.ID.@integer = CONNECT ( "trl", "Excel Potline" )
THEN READ ( trl.ID.@integer, "trl.r18.@f" )
THEN READ ( trl.ID.@integer, "trl.r34.@f" )
THEN line.current.@float   = trl.r18.@float
THEN line.voltage.@float   = trl.r34.@float
THEN trc.TYPE.@string is "DDE"
THEN trc.MAPFILE.@string is "D:\comdale\exp_1.mpf"
THEN trc.ID.@integer = CONNECT ( "trc", STRCONCAT ( "Excel Cell_",
cell.selected.@string ) )
THEN READ ( trc.ID.@integer, "trc.r14.@f" )
THEN READ ( trc.ID.@integer, "trc.r15.@f" )
THEN READ ( trc.ID.@integer, "trc.r17.@f" )
THEN READ ( trc.ID.@integer, "trc.r20.@f" )
THEN READ ( trc.ID.@integer, "trc.r22.@f" )
THEN READ ( trc.ID.@integer, "trc.r23.@f" )
THEN READ ( trc.ID.@integer, "trc.r24.@f" )
THEN READ ( trc.ID.@integer, "trc.r40.@f" )
THEN READ ( trc.ID.@integer, "trc.r50.@f" )
THEN READ ( trc.ID.@integer, "trc.r51.@f" )
THEN READ ( trc.ID.@integer, "trc.r52.@f" )
THEN READ ( trc.ID.@integer, "trc.r53.@f" )
THEN READ ( trc.ID.@integer, "trc.r55.@f" )
THEN cell.voltage_fluctuation.@float   = trc.r15.@float
THEN cell.voltage_nominal.@float   = trc.r17.@float
THEN cell.voltage_average_day.@float   = trc.r14.@float
THEN cell.voltage_deviation.@float   = ABS ( cell.voltage_nominal.@float   -
cell.voltage_average_day.@float   )
THEN anode_effect.number.@float   = trc.r20.@float
THEN anode_effect.voltage_average.@float   = trc.r22.@float
THEN anode_effect.voltage_max.@float   = trc.r23.@float
THEN anode_effect.duration.@float   = trc.r24.@float
THEN anode.adjust_number.@float   = trc.r40.@float
THEN cathode.voltage_drop.@float   = trc.r50.@float
THEN metal.weight_tapped.@float   = trc.r51.@float
THEN metal.level.@float   = trc.r52.@float
THEN bath.level.@float   = trc.r53.@float
THEN bath.temp.@float   = trc.r55.@float
THEN GOTO ( "alarm2" )
EndRule
```

In this alarm rule example, the on-line process data connection function and each monitored data are formatted. That is the inlet of the alarm module, which connects the directly measured or already stored process data. Follows the "THEN GOTO ("alarm2")" command, the other rule of the alarm module will continue to gather another kind of process information, which is mainly concerned with the manually measured data.

A-3.1.2 Example rule of process data status detection

Depending upon the value of the received on-line process data, the next task of the alarm module is to detect the status of these data. The following rule is an example to detect the alarm status (See Table A-3. 2).

Table A-3. 2 Rule alarm2

Rule ***@name = alarm3_12_2*** ***@priority = 5***
If iron.impurity.@s is "ABNORMAL" *AND iron.impurity.@f >=0.170000* *AND iron.impurity.@f <0.190000* *THEN iron.impurity.@s is "HIGH"* *THEN raw-material_1.check.@s is "Contents of the relative raw materials."* *THEN raw_material_2.check.@s is "1). Alumina chemical composition [F_e, S_i etc.]"* *THEN raw_material_2.check.@s is "2). Ash content in anode material"* *EndRule*

The rule uses the specific alarm limitation to detect that the value of iron impurity and converts it to a fuzzy expression "HIGH", which is acceptable for the general modules. Depending upon this alarm, an announcement is released for the operator and the relevant detections will be continued. All the information of these detections and conversions from the alarm modules will be sent to the general knowledge modules for the further reasoning. These two examples show that the external information is only concerned with the alarm module. Therefore, this improved structure of knowledge base will benefit the modification and updating works.

A-3.2 Utilizing uncertainty expression to improve the reasoning process

In real process, a substantial number of the information presents the uncertainty. To correctly treat the uncertainty information is the prerequisite condition of the reasoning. The Degree of Certainty (DC) is applied in the present system to improve the ability of uncertainty information treatment. In the certain rules, the conclusion statements may also be attached with a Certainty Factor (CF). This CF indicates the confidence in that conclusion statement. The DC is used for describing the uncertainty of the symptom, but the CF is used for describing the uncertainty of the conclusion.

Here is an example to show the improvement of diagnosis process by using CF. In this example; two testing cases of "High iron impurity" sub-system of

ENGES are compared. In case 1, the CF was not concerned to the certain rule. But in the case 2, the CF was an additional consideration. The result of the reasoning process of these two cases presents some differences.

In the case 1, at first the inference engine fires a rule to release a question to ask for the value of the iron content in the metal pad. If the value is less than 0.2%, then the first conclusion will be judged as: "The iron content in the metal pad is normal." Considering some special cases, there may be some potential problems, which still affect the iron content, even the value of iron content in a normal range. So another rule was designed to search for this potential possibility. The consequent rules will be fired for gathering more information. As in case 1 the DC is not used to describe the certainty of the fact and the CF is also not attached with the relevant conclusion. Therefore, all the corresponding rules will be executed and all the relevant questions will be asked. The questions, answers and conclusions in this procedure are listed as:

1. "What is the value of the iron content (%) in metal pad?"
2. Diagnosis Conclusion: "The iron content in the metal pad is normal," when typing value is 0.09%.
3. "Is the iron impurity increase in metal pad increased during last analysis period?"
4. Then given the diagnosis report: "The iron impurity in metal pad will be abnormal," when the answer is "Yes."
5. "Is the anode stem immersed in the bath?"
6. "Did you find out some iron accidental contamination the metal?"
7. "Did you find out some cathodes failure?"
8. "Does the iron impurity continuously increase which is caused by the cathode collector bar?"

But in real process, when the conclusion of "The iron impurity is low" with high confidence, then there is no need to ask the consequent questions and the considered potential problems could be omitted, as well the whole reasoning process also could be simplified. For this reason, in case 2 the CF was considered to describe the confidence of the conclusion of rules. For example, the different CFs were attached to the conclusion "The iron impurity is normal." If the iron impurity is less than 0.1% and the CF=90.0, but if the range is from 0.1% to 0.2%, the CF=75.0. And then, all the facts were described with the estimated DC. Based on such improvements, the corresponding reasoning process is simplified.

For example, if the iron impurity is also "0.09%", then the consequent questions and relevant CF are as follows:

1. Conclusion: "The iron impurity in metal pad is normal."
2. The CF of "The iron impurity is normal." is 90.0.
3. Terminating the diagnosis process depends on the answered information.

As when the CF of conclusion "Iron impurity is normal" is 90.0, that means this conclusion is with high confidence. Therefore, there is no need to doubt it, as well as to ask another question. When the iron impurity is in the range from 0.1% to 0.2%, then the CF of the conclusion is reduced to 75.0. This reduced confidence will lead the inference engine to release corresponding questions to gather further information to detect whether a potential problem exists, which could lead to increasing the iron impurity.

The comparison of these two cases shows a fact: using the numerical expression of the uncertainty could realistically reflect the complexity in practice and could help to improve the reasoning process.

A-3.3 Improve the diagnosis speed

The speed of diagnosis process is one of the performance indexes of an expert system. To improve the diagnosis speed, two steps should be adopted:

- Deeply understand the acquired knowledge
- Improve the design of corresponding rule base and select the correct searching strategy

The diagnosis speed is concerned with two concepts: the searching time and the diagnosis time. The searching time is the time of scanning all the rules, facts, on-line data and available information. The diagnosis time is concerned to the time of the question release, the required information typing and the time to do the consequent diagnosis.

To determine the searching time, several tests were executed to compare the variance between the different scales of systems. Three sub-systems are selected for the testing, the numbers of the rule applied in these sub-systems are as: “Sludge accumulation” (14 rules), “Excessive anode effect number” (35rules) and “OPEES-1” (298 rules). The searching times are roughly:

"Sludge accumulation" sub-system: less than three seconds
"Excessive anode effect number" sub-system: less than four seconds
"OPEES-1" sub-system: less than eight seconds

The test results show that all the searching times are less than eight seconds whether the rules base contains twenty or three hundred rules. It indicates a fact that only seconds in time difference is not an important factor to affect the diagnosis speed, so it could be considered that search time does not matter much with the number of rules applied, especially on such scale of the rule base, which consisted of several hundred rules. However, another test result shows that most time is expended in the question answering or the required value typing, as well as the diagnosis time (See **APPENDIX A-4.1 "Diagnosis time for different number of cells."**) That means more rules applied, then more chaining questions will be released, more answers are needed to type and more time will be expended to do the relevant diagnosis. Thus, to improve the diagnosis speed will focus attention on the improvement of the rule base (to reduce the number of questions) and the selection of an appropriate searching strategy (to search for an efficient searching strategy).

Therefore, the second step of the diagnosis speed improvement is concerned with the rule base design and the searching strategy selection. To improve the rule base, deeply understanding of the corresponding knowledge is the primary goal. Truly following the thinking logic of human expert will help the inference engine avoid undue complex search links, reduce unnecessary questions and could make a smart decision. The major approach of improvement is to determine the correctly chaining link. Based on the obtained facts to point out the key problem is very important for the next searching process.

Here is an example of the comparison of two test results of "Excessive number of anode effects" sub-system. In case 1, the search strategy is set as "Search all rules." When the number of anode effects is detected as "2", which is higher than normal value, the inference engine will fire all relevant rules to release the successive questions:

1. Are there some alumina bins empty?
2. Are there some alumina bins obstructed?
3. Are there some alumina bins gate obstructed?
4. Are there some alumina point feeder obstructed?
5. Are there some alumina point feeder holes obstructed?
6. Is the crust very hard and more difficult to set anodes?
7. Is the alumina thermal conductivity increased?
8. Is the alumina dissolution decreased?
9. Is the fluorocarbon emission increased?
10. Is the bath level low?
11. What is the value of anode effect voltage?
12. Is bath temperature low?
13. Is anode current distribution abnormal?

As seen above, if the number of anode effect is high, there are 13 questions to be asked. Could the diagnosis speed of this problem be improved? To answer this question, the primary step should be done is to understand the deeper causes of the problem. However, an experienced engineer could find some of these questions are not needed to ask due to the analysis of the individual case.

Therefore, follow the thinking logic of human expert to improve the chaining link of the rule base could avoid the invalid questions and speed up the diagnosis process.

In the case 2, the actual reason of anode effect was analyzed, and then the chaining link of the rules was redesigned. The searching strategy was also changed as: "Terminate when no new sub goal found in successive rule." For example, when the alumina bins are found empty, then there needs no checking the obstruction of the feeding system and the change of alumina properties. This analysis results will be counted as chaining fact in the redesigned rule base. The test result of this improved rule base shows that required questions in the diagnosis process will be reduced to only six:

1. Are there some alumina bins empty?
2. Is the crust very hard and more difficult to set anodes?
3. Is the bath level low?
4. What is the value of anode effect voltage?
5. Is bath temperature low?
6. Is anode current distribution abnormal?

By this way, the final diagnosis result is still the same but the invalid questions are avoided and the expended time in the diagnosis process is reduced to 50%. More importantly, some unnecessary inspections of the feeding system and the alumina properties in the real smelter could be avoided. Obviously, using such approach to improve the rule base will be more beneficial to the larger scale rule base, where more expended time could be saved as well as to quicken the entire diagnosis speed.

A-3.4 Utilizing multiple symptoms criterion to improve the reasoning process

Actually, most of the problems occurred in the production are accompanied with the multiple symptoms. The treatment of the multiple symptoms is one of the goals of rule base design. As described in **Section 6.5.3 "Multiple symptoms criterion,"** the multiple symptoms criterion is an important step to improve the treatment ability to the real complex information. The further test and discussion is described in the following.

In order to improve the present system, the existing expert system was studied and some problems were found in the multiple symptoms cases. These problems were concerned with the analysis of the symptoms and the structure of the rules. Because in the knowledge acquired from the domain experts, the multiple symptoms of a specific case are normally only listed as such, there is no indication of the rank of these symptoms to be concerned. If these symptoms are coded into the rules base as such, as well as are arranged by the logic **AND**, some false results could be obtained. The following example (See Table A-3. 3) shows how the problem was occurred:

Table A-3. 3 Rule alumina property1

Rule
@name = alumina_property1

IF anode_effect.number.higher
AND bath.temperature.too_low
AND crust.hardness.higher
AND cell.voltage.unstable
THEN alumina.property.abnormal is TRUE
THEN TEXT ("Diagnosis Report:
Alumina properties look abnormal! ")
THEN TEXT ("Suggestion:
Go to the laboratory to check the analysis results.")
EndRule

There are four symptoms listed in the premise part of this rule. The ranks of these symptoms follow the description in the required knowledge without any analysis. That means all the symptoms are considered with identical importance and CF, and they will be treated with the same priority. If any symptom does not occur or the operator omits some relevant observation, the final conclusion will be false, even if the remainder symptoms of the problem have occurred. For example, if the symptoms of the *"cell.voltage.unstable"*, the *"anode_effect.number.higher"* and the *"bath.temperature.too_low"* (they are the major monitored process data) are confirmed, only the *"crust.hardness.higher"* is not satisfied, the conclusion: *"alumina.property.abnormal"* will be also FALSE. The hardness of the crust is a manual measured parameter, which is often accompanied with an uncertainty. If there is no any attached CF, this information will easy lead to a false results in the reasoning process. To avoid this kind of problem, two steps should be adopted:

- Analysis of the multiple symptoms of each problem, to identify the differences between them and searching for the essential cause of each symptom.
- Improve the structure of the rules; rearrange the multiple symptoms by a reasonable logic, which could really represent the knowledge of the experts.

For example, in this particular rule, four symptoms were analyzed and they are rearranged into two premise parts of the rule: the symptoms of the *"anode_effect.number.higher"* and the *"cell.voltage.unstable"* could be considered to be caused by the same reason and the *"cell.voltage.unstable"* is also considered as another representation of the *"anode_effect.number.higher".* Therefore, they are related by the logic **OR**. The remains of two symptoms: *"bath.temperature.too_low"* and *"crust.hardness.higher"* will be also treated by the same method.

Finally these four symptoms are rearranged by the different logic in the rule, and the result shows some differences. (See Table A-3. 4.)

Table A-3. 4 Improved rule alumina property1

```
Rule
@name = alumina_property1

IF  anode_effect.number.higher | cell.voltage.unstable
AND  bath.temperature.too_low | crust.hardness.higher
THEN alumina.property.abnormal is TRUE
THEN TEXT ( "Diagnosis Report:
Alumina properties look abnormal! ")
THEN TEXT ( "Suggestion:
Go to the laboratory to check the analysis results.")
EndRule
```

By using this modified rule, when the *"crust.hardness.higher"* is also not satisfied, the conclusion becomes *"alumina.property.abnormal is TRUE."* Why is there a different result obtained under the same condition? The basic reason is that the symptoms were reset by the different logic, which could meet the essential thinking of the experts. In the second part of the premise of this rule, the *"crust.hardness.higher"* was considered as another representation of the symptom of *"bath.temperature.too_low"*. These two symptoms are also related by the logic **OR** and are caused by the same reason. When the symptom *"bath.temperature.too_low"* was confirmed, that could be considered being equal to the *"crust.hardness.higher"* to a certain extent. Therefore, in spite of the *"crust.hardness.higher"* was not satisfied by some reasons (such as a faulty measurement or an error message,) the conclusion was still correct. In this way, the structure of the rules could reflect the real substance of the knowledge of the human experts. The reasoning process can be improved to avoid some unreasonable reasoning results.

A-3.5 Conclusions

During the system development, several cases were studied to verify the results under different conditions, and the different search strategies were tested to observe the different efficiency of the reasoning processes. Depending upon these testing results the conclusions are summarized as follows:

A-3.5.1 Knowledge - acquisition and study

During the knowledge acquisition, one of the basic principles is always to remember that the knowledge will be used for the real application. This means the expected knowledge will not only be gathered piece by piece, but more importantly, the potential relationship between them will have to be found. Such kind of knowledge could present its substance, and will reflect the train of thought of experts. The best way to obtain such thorough knowledge is by interviewing experts, but they are not always available. Therefore, after the knowledge has been acquired, the further review of the knowledge becomes an important responsibility for the knowledge engineer. The study of the knowledge in this stage is concerned with two aspects:

- Domain knowledge acquaintance
- Knowledge rearrangement

The former is the further study of the entire knowledge about the concerned domain. The latter is based on the understanding of the knowledge and its conversion into a formal logic to meet the expert system requirement. This logic will follow the thinking process of an expert, as well as give the expert system an ability to solve the real problems like human expert.

A-3.5.2 Applicability of expert system

No doubt, any knowledge engineer will desire to see whether its expert system can be readily applied. To realize this target, one of the basic requirements is the applicability of the expert system. Two relevant aspects should be considered: the knowledge base must contain enough knowledge to cover the particular application and the knowledge base should be easy to modify to suit the various situations. The former is considered as the generality of acquired knowledge, but the latter is concerned with the particular knowledge and the technique of knowledge base design. How to ingeniously separate the general and particular knowledge and let them work in a coordinated matter is important for the system applicability. The modular rule base is one of the better choices, which has been proved by the present system. It provides some capability to allow the user to easily update the rule base to suit the different situations.

A-3.5.3 The tolerance for uncertain or unclear information

In practice, the system cannot always work in an ideal condition; in most cases the system will face many uncertain, unclear or even erroneous messages. To reduce the negative effect of these undesired messages is an important consideration for the system design. Allowing the system to have some tolerance for such kind of information could improve the working reliability of the system.

This tolerance will allow the user using some uncertain or inaccurate information to do the reasoning but to avoid affecting the correctness of the result.

A-3.5.4 Speed and correctness

Based on the analysis of the testing results above, for small or regular size rule bases, to avoid the invalid questions and to improve the correctness of final result, is the primary consideration of the system design. Too many invalid questions will affect the speed of the diagnosis process. To ensure a reasonable chaining linked rule base and to obtain a correct result, three aspects should be considered: the knowledge acquisition, the required process information and the knowledge base design. The last two aspects are often concerned with more relatively varied factors and therefore need careful handling.

APPENDIX 4 CASE STUDY – TEST AND ANALYSIS

In order to evaluate the present expert system, several tests were done. These testes were concerned with the following aspects:

- To evaluate the performance when the system works in a real situation
- To evaluate the effect of the different alarm status
- To evaluate the effect of uncertain information
- To find out the limitation of the operational performance
- To confirm the available range of the relevant parameters.

A-4.1 Diagnosis time for different number of cells

The diagnosis time of the present system has been discussed in **APPENDIX A-3.3 "Improve the diagnosis speed,"** where only one cell was concerned. For the real application, the system must be designed for all cells of a smelter. Normally, there are 100 – 200 cells in one smelter. In this situation, the diagnosis time for hundreds of cells becomes a serious consideration. Obviously, too long time for diagnosis would make no sense in a continuously changing situation. To obtain numerical evaluation of the performance of the OPEES-2, nine cases were selected for testing purpose, which are listed in Table A-4. 1. In order to emulate the real diagnosis process, three situations are considered: 5%, 10% and 15% cells are in the abnormal status. Each situation includes three cases, which are concerned with one, two and three alarm data occurrences.

Table A-4. 1 Diagnosis time testing

Diagnosis time testing			
1	5% cells are in abnormal status (10 cells)	Case 1.1	1 alarm data
		Case 1.1	2 alarm data
		Case 1.3	3 alarm data
2	10% cells are in abnormal status (20 cells)	Case 2.1	1 alarm data
		Case 2.2	2 alarm data
		Case 2.3	3 alarm data
3	15% cells are in abnormal status (30 cells)	Case 3.1	1 alarm data
		Case 3.2	2 alarm data
		Case 3.3	3 alarm data

The diagnosis time counting is based on single cell diagnosis process. Actually, the diagnosis time is a relative index of performance; it would be varied depending upon the different skill level of the users and the accuracy of the relevant information provided. But it can still be considered as a valuable parameter to indicate the performance of the system.

As the different alarm data are concerned with the different problems, which link to the different rules sets and so the relevant diagnosis times are also different. Therefore it only can be represented by an average value of the diagnosis times of several alarm data. The following three typical data were selected to count the diagnosis times and then to obtain an average diagnosis time from them:

1. Number of anode effects
2. Iron impurity content in the metal
3. Bath temperature.

The OPEES-2 was selected for testing as it was designed to match the measurement system of the current smelter, where only limited process data can be measured continually, but most process data should be provided manually (See **Section 6.3.2.2 "Structure of OPEES-2"**). To reduce the effect of different manual operations, the diagnosis time is counted under the following conditions:

- All the required information is ready
- The operating behavior is normal
- *The cell operating status is abnormal but with only few alarmed process* data occurred.

The diagnosis time includes two parts: the searching time for the on-line data file and the consequent diagnosis time. As the searching time is less than 5 seconds, it could be ignored and only the diagnosis time is counted. The testing results are shown in Table A-4. 2:

Table A-4. 2 Diagnosis time of typical alarm data

Diagnosis time of typical alarm data (M:S)		
1	Number of anode effect is high	1:35
2	Iron impurity content in the metal is high	1:45
3	Bath temperature is high	1:50

As these diagnosis times are counted independently from each data, all the time about the information typing and diagnosis process are counted respectively. Actually, for one cell with several alarm data, all the required information only needs to be typed once. Therefore, for more alarm data only need to add the reasoning times. The diagnosis time of three data and the total time are recounted as in the list in Table A-4. 3. It should be indicated that the diagnosis time of the first data is counted on both the information type time and its diagnosis time.

Table A-4. 3 Diagnosis time of three alarm data of one cell

Diagnosis time of three alarm data of one cell (M:S)		
1	Number of anode effect is high	1:35
2	Iron impurity content in the metal is high	+ 0:20
3	Bath temperature is high	+ 0:25
	Total diagnosis time of three data	**2:20**

Based on this consideration, more data have been tested and the average times of different number of the alarmed data are listed in Table A-4. 4.

Table A-4. 4 Average diagnosis time of one cell

Average diagnosis time of one cell (M:S)	
1 alarm data	1:35
2 alarm data	2:00
3 alarm data	2:20

Depending on the average diagnosis times of these alarm data, the corresponding diagnosis time of different number cells are counted as per Table A-4. 5. In this testing case, there is a total of 200 cells to be monitored and three situations are considered: 5%, 10% and 15% of the cells are in the abnormal status. Therefore, the corresponding numbers of abnormal cells are 10, 20 and 30. Each situation includes three cases, which are concerned with one, two and three alarm data occurrences.

Table A-4. 5 Diagnosis time comparison

Diagnosis time comparison (M:S)		
1 cell is in abnormal status	1 alarm data	1:35
	2 alarm data	2:00
	3 alarm data	2:20
5% cells are in abnormal status (10 cells)	1 alarm data	15:50
	2 alarm data	20:00
	3 alarm data	23:20
10% cells are in abnormal status (20 cells)	1 alarm data	31:40
	2 alarm data	40:00
	3 alarm data	46:40
15% cells are in abnormal status (30 cells)	1 alarm data	47:30
	2 alarm data	60:00
	3 alarm data	70:00

It is obvious that diagnosis times have to be increased when the number of abnormal cells is increased. If the data file refresh cycle is set to 30 minutes, diagnosis times longer than 30 minutes make no sense. To solve this problem, three approaches are discussed in the next section.

A-4.1.1 Discussion

Monitor cycle and diagnosis time

Using OPEES-2 to monitor the cell status, two kinds of the process data are considered: one is the on-line data, which can be measured continually and the remainder, which can only be measured manually. To choose an appropriate monitor cycle is important to ensure that the abnormal status can be diagnosed in the right amount of time. This cycle should consider both the diagnosis time of the expert system and the time of the cell status variation. Because many process data of the cell are slowly varied, for example, the ledge profiles, the bath level, the weight of tapped metal and impurity content of metal. Thus, the data file refresh cycle could be set as 30 minutes, which means that every 30 minutes all of the process data, which are saved in the data file will be refreshed at least one time. Actually, the on-line data refresh cycle can be faster (less than five minutes), but remainder of the process data refresh cycle is as slow as 30 minutes.

This is no doubt that the 30 minutes monitor cycle is too slow for the on-line data, which varies relatively fast. As OPEES-2 can cooperate with the controller of each cell, any variation of the on-line data can be adjusted immediately. Therefore, the relatively slow diagnosis process should not affect the adjustment of the on-line data. However the problem is that when the number of abnormal cells is increased, the diagnosis times will be too long. For example, when the number of the abnormal cell increases to 20, the diagnosis time for three problems of every cell will be 46 minutes and 40 seconds, a time that is not acceptable for the on-line diagnosis. To solve this problem, more detailed discussion is represented in the next part.

Distributed structure

In the normal production, 10% of cells with alarmed data is a reasonable number, which means in a monitor cycle, OPEES-2 should finish the diagnosis tasks of at least 20 cells. According to the testing results in Table A-4. 5, the appropriate number of cells to be diagnosed in one monitor cycle is only 10 cells. To solve the problem of too long of a diagnosis time, one of the efficiency methods is to use more OPEES-2 sub-systems to construct a distributed system. In this system each OPEES-2 is only in charge of 50 cells. Four identical OPEES-2 sub-systems can be applied for a total of 200 cells (See Figure A-4. 1).

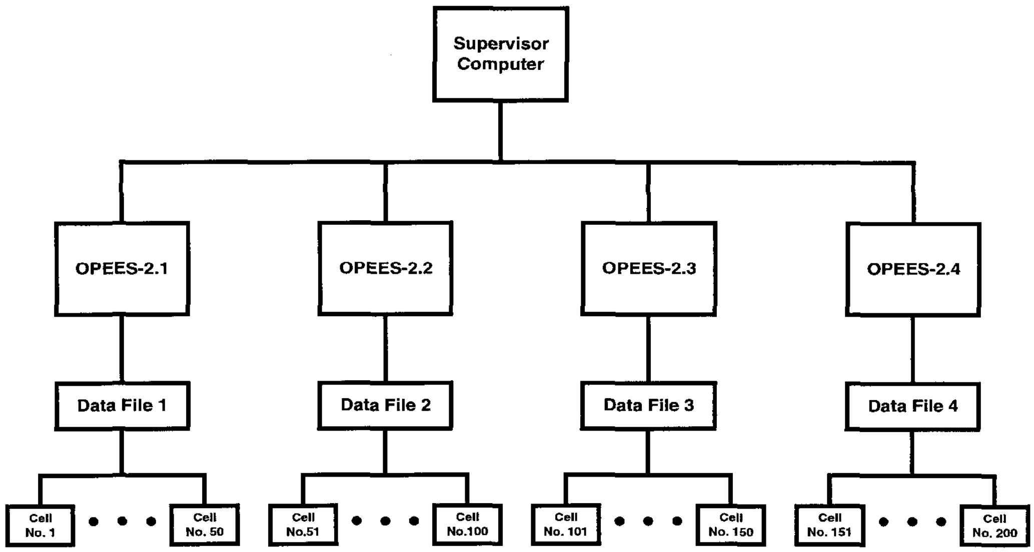

Figure A-4. 1 Distributed structure of the improved OPEES-2

In this distributed system, OPEES-2 does not need to modify its knowledge base and structure, and it can still be in charge of 10 cells in a monitor cycle. Thus, four OPEES-2 sub-systems could work together in the meantime and a total of 40 abnormal cells could be diagnosed in a monitor cycle. Therefore, the percentage of the available abnormal cells to be diagnosed is increased to 20%. Although this proposed system couldn't reduce the manual measurement time and the operating time, it is still an available approach to improve the diagnosis time.

OPEES-2 improvement

Another possible approach to speed up the diagnosis time of OPEES-2 is concerned with the modification of its rule base. In the present system, each alarmed data will be treated with same priority. Actually, the alarmed data should represent different alarm statuses, as well as with the different deviation. For instance, the alarmed data could be expressed in five levels: "Very High," "High," "Normal," "Low," and "Very Low" depends on its deviation (See **Section "6.4.3.2 Module hierarchy of knowledge base"**). If the two extreme levels ("Very High" and "Very Low") are bestowed with higher priority and the "High", "Low" are bestowed with lower priority, then OPEES-2 will do the prior diagnosis for the alarmed data with higher priority in all instances, then the lower priority. By this way, the system gives first priority to the more serious alarmed data and could find out the causes of the problem in time. Therefore, the effect of the abnormal status could be reduced in a monitor cycle. The treatment of the priority of the alarmed data is illustrated in Figure A-4. 2.

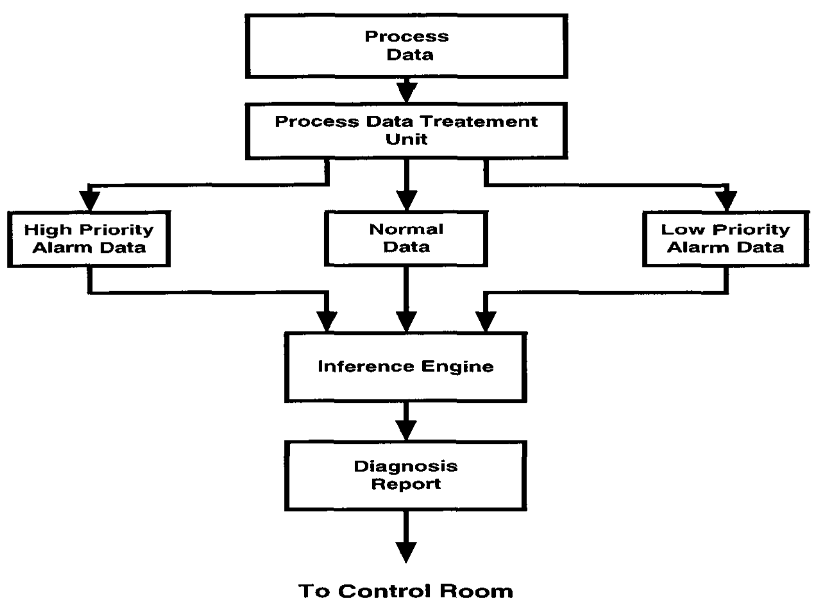

Figure A-4. 2 Higher priority alarm data treatment process

There the additional process data treatment unit will scan all the required information either of on-line data or manually typed data to find out the abnormal data, and then assigns the different priorities to them depending upon their variations. Then the inference engine will follow the rank of the priorities to diagnose the causes of the alarmed data.

Improving the measurement system

To improve the diagnosis speed, the thorough approach is to improve the existing process data measurement. Most diagnosis time is spent on the process data measuring and typing. Improvement of measuring methods and devices will reduce the diagnosis time. The on-line diagnosis could be really possible only when this basic problem will have been solved.

A-4.2 Diagnosis testing with different uncertainty information

To diagnose the problem in the real cell operation, the degree of certainty (DC) is often used to represent the uncertainty information and incomplete measurements. With different DCs, the reasoning results could be quite different. In order to find out the effect of the different DCs of the present system, two cases were studied. In case 1, four sets of DCs with different values were selected to test three problems respectively. In case 2, five sets of different values of DCs were selected to test the multi-symptom problems.

A-4.2.1 Effect of different DC values on problem diagnosis

In the reasoning process, large amount of information is gathered through the question interface. Different DC could be assigned at this level. To test the effect of different DC values on the final diagnosis results, four different values of DC were selected for three problems in ENGES. The values of the DC and the problems are listed in Table A-4. 6.

Table A-4. 6 Testing values of DC and problems

Tested value of DC		Tested Problems
1	DC=100%	1. Excessive number of anode effect 2. Higher Iron impurity content in the metal 3. Anode carbon quality
2	DC=75%	
3	DC=50%	
4	DC=35%	

For each problem, four tests were executed by using each set of DC. The results are listed in Table A-4. 7.

Table A-4. 7 Testing results of different DC

		Diagnosis results		
Case No.	**DC value**	**Problem 1**	**Problem 2**	**Problem 3**
Case1	DC=100%	Success	Success	Success
Case 2	DC=75%	Success	Success	Success
Case 3	DC=50%	Success	Success	Success
Case 4	DC=35%	Failure	Failure	Failure

In case 1 – case 3, the reasoning processes were successful and the final results were also correct. Why could the different DCs lead to the same results? The reason is that the threshold value of DC is set as 50% by default, therefore any values which are more than 50% will be concluded that the premise of this problem is "True." Thus, the same information with different DC (more than 50%) will be treated as equal.

In case 4, as the DC is less than 50%, the inference engine will interrupt the diagnosis process because all the DC of the premises of the rules were not satisfied. A final report will indicate that the system cannot find the problem depending on the supplied information with too small DC value.

During the reasoning process, based on the typed DC value, another parameter NDT (Net degree of truth) will be calculated, this is an internal parameter, which indicates the truth of the conclusion of each rule. The NDT is calculated when all premises of a rule are examined and will depend on the different DC of each premise. As mentioned above, using the premise with different DC value (more than 50%) could get an identical diagnosis result. That means that although the results are the same, the truths of the results (NDT) may be different. Unfortunately, the present system cannot supply this parameter in the diagnosis report, as the NDT is only calculated in the background by the system. In order to indicate the effect of the different DCs and show confidence in the results, it is important to provide a similar parameter for further improvement.

An example of the relationship between DC and NDT of each rule of case2 (Problem 1) is listed in Table A-4. 8. The average NDT of all rules is 82.95%. This result indicates the fact that if the DC value is higher than the defaulted threshold value, the final result still has a higher confidence.

Table A-4. 8 DCs and NDTs of case 2 (Problem 1)

DCs and NDTs of case 2 (Problem 1)		
DC value	**Rule No.**	**NDT %**
DC=75%	1	100
DC=75%	2	75
DC=75%	3	75
DC=75%	4	75
DC=75%	5	75

DC=75%	6	100
DC=75%	7	75
DC=75%	8	100
DC=75%	9	75
DC=75%	10	100
DC=75%	11	75
DC=75%	12	75
DC=75%	13	75
DC=75%	14	100
DC=75%	15	75
DC=75%	16	100
DC=75%	17	75
DC=75%	18	100
DC=75%	19	75
DC=75%	20	75
DC=75%	21	75
DC=75%	22	75
Average NDT (%)	**82.95**	

In case 1 – case 4, the fixed values of DC are tested to do the diagnosis. Using such a relatively simple method can easily find the relationship between different values of DC and their final results. However, this method does not reflect the complexity of the real world. Different percentages of the DC applied for multi-symptoms are studied in next section.

A-4.2.2 Effect of different DCs selected for multi-symptoms

In this set of cases, the different DC values are selected for both major and minor symptoms as well as tested with three problems. The values of DCs are selected in different percentages, which are listed in Table A-4. 9.

Table A-4. 9 Testing DC sets for multi-symptoms

Case No.		Major symptom				Minor symptom			
	DC value	100%	75%	50%	35%	100%	75%	50%	35%
1	%	80	10	5	5	80	10	5	5
2		65	20	10	5	65	20	10	5
3		50	20	20	10	50	20	20	10
4		30	20	20	30	30	20	20	30
5		15	10	35	40	15	10	35	40

The different sets of percentages of DC are selected to emulate the different situations of the real diagnosis process. In case 1 – case 3, the percentage of the information with DC=100% in both major and minor symptoms are 80%, 65% and 50% respectively as well as the degree of certainty is in a downward trend. The DC values of the remainder information are selected as DC=75%, 50% and 35%, which occupy 5% - 20% respectively. In case 4 and case 5, the information with DC=100% only occupies 30% and 15% respectively, which emulates a case of lower confidence information. The numbers of the rules associated with these three problems are provided in Table A-4. 10 and the rule number distributions of five cases are listed in Table A-4. 11. The testing results are shown in Table A-4. 12.

Table A-4. 10 Rule numbers of three problems

Rule numbers	
Problem 1	35 rules
Problem 2	13 rules
Problem 3	9 rules

Table A-4. 11 Rule distribution of different DC values

	DC values (%)	Problem 1	Problem 2	Problem 3
Case1	DC=100%	28 rules	10 rules	7 rules
	DC=75%	3 rules	2 rules	1 rules
	DC=50%	2 rules	1 rules	1 rules
	DC=35%	2 rules	0 rules	0 rules
Case 2	DC=100%	24 rules	9 rules	6 rules
	DC=75%	4 rules	2 rules	1 rules
	DC=50%	4 rules	1 rules	1 rules
	DC=35%	3 rules	1 rules	1 rules
Case 3	DC=100%	21 rules	8 rules	5 rules
	DC=75%	7 rules	2 rules	2 rules
	DC=50%	4 rules	2 rules	1 rules
	DC=35%	3 rules	1 rules	1 rules
Case 4	DC=100%	17 rules	6 rules	4 rules
	DC=75%	10 rules	4 rules	3 rules
	DC=50%	7 rules	2 rules	1 rules
	DC=35%	1 rules	1 rules	1 rules
Case 5	DC=100%	14 rules	5 rules	4 rules
	DC=75%	10 rules	4 rules	2 rules
	DC=50%	8 rules	2 rules	2 rules
	DC=35%	3 rules	2 rules	1 rules

Table A-4. 12 Testing results for multi-symptoms

Testing results for multi-symptoms						
Case No.	Problem 1		Problem 2		Problem 3	
1	Success	Average NDT=90.8 %	Success	Average NDT=90.6 %	Success	Average NDT=95.8%
2	Success	Average NDT=85%	Success	Average NDT=87.5 %	Success	Average NDT=87.5%
3	Success	Average NDT=73%	Success	Average NDT=71.8 %	Success	Average NDT=70.8%
4	Success	Average NDT=57%	Failure	Average NDT=49%	Failure	Average NDT=37.5%
5	Failure	Average NDT=40%	Failure	Average NDT=40.6 %	Failure	Average NDT=33.3%

In case1 – case 4, most results are successful except the last two, but the average DNT of each problem ranges from 90.8% down to 57%. In case 5, too low DC values lead to the interruption of the diagnosis process and the NDT is also lower.

A-4.2.3 Discussion

Upon examination of the tested results, four points are summarized as follows:

DC value and threshold value

It is found that different DC value could still lead to the same correct final results. The main reason is the defaulted threshold value. If the DC values are higher than the threshold value, the diagnosis results can be identical. The NDT of each result nevertheless varies depending on its DC value. In the present system, the defaulted threshold value is 50%, so any information with the DC value higher than this value will not affect the correctness of final results.

Result and rule set size

For different problems, the sizes of the rule sets are different. It is found that the amount of information with lower DC values and the number of rules could affect the final results. If a DC value changes from higher to lower than threshold value, the difference in the percentage of NDT will be more obvious in a smaller size rule set. For example, there are only 9 rules applied to problem 3. In case 3, if one more premise is changed form DC=100% to DC= 35%, the NDT will be changed from 70.8% to 54%. The NDT is decreased by 16.8%. In the same case but of problem 1 (with 35 rules), the DC of one premise is changed from DC=100% to 35%. The NDT will change from 71% to 68.8%, by 2.2% only. Therefore, in a smaller size rule set, the lower DC value must be treated very carefully. The effect of the uncertainty information will obviously affect the final results more than the larger size rule set.

Effect of the lower DC in the initial stage of diagnosis

If the DC value is too low in the initial stage of the diagnosis process, the reasoning process and the accuracy of final result may be affected or even the diagnosis process may be interrupted. In the initial stage, the fired rules are normally with higher priorities. Any information with too low DC value will seriously affect the first few steps of searching chain links that may even lead to a wrong linking direction. For example, in the problem 1 (See **Figure A-6. 6 Different priorities of the rules**), the first question is: "What is the number of anode effects during the last 24 hours?" If the DC of typed information is less than 50% and with for the consequent question: "What is the value of the anode effect voltage?" then the diagnosis process could be interrupted. The NDT was also shown as too low. This fact indicates that the information of the first stage must be with a higher DC value in order to take a correct searching direction.

Accurate information and NDT

The tests clearly show that the accuracy of the required information is important to ensure the success of the diagnosis process. To satisfy this requirement, two improvements should be made:

- Improve the measurement technology and provide the timely on-line data.
- A warning massage could be added into the system detection function module. When the information with too low DC value is typed in, the user would be asked to reconfirm the DC and the software would show the corresponding NDT.

APPENDIX 5 FLOWCHART OF AEPES OPERATION

To help understand how AEPES works, Figure A-5. 1 gives an entire flowchart of the system operation. The operating procedure is:

- Start the system
- Select one of three sub-systems
- Follow the instructions to select desired line, cell or problem
- Answer questions with the required information or type the required values of the data
- When reasoning process is finished, display the diagnosis report
- Follow the suggested actions to solve the problem
- Make more selections to continue further diagnosis
- If there are no more questions, terminate the system.

When the system runs in the on-line mode, several preparatory steps should be carried out:

- Communication Bridge will be set in both the internal system and the communication network.
- Relevant data file will be set in both the internal system and the communication network.

To ensure the correctness of reasoning process, all the questions should be answered and the required value of data should be in reasonable ranges.

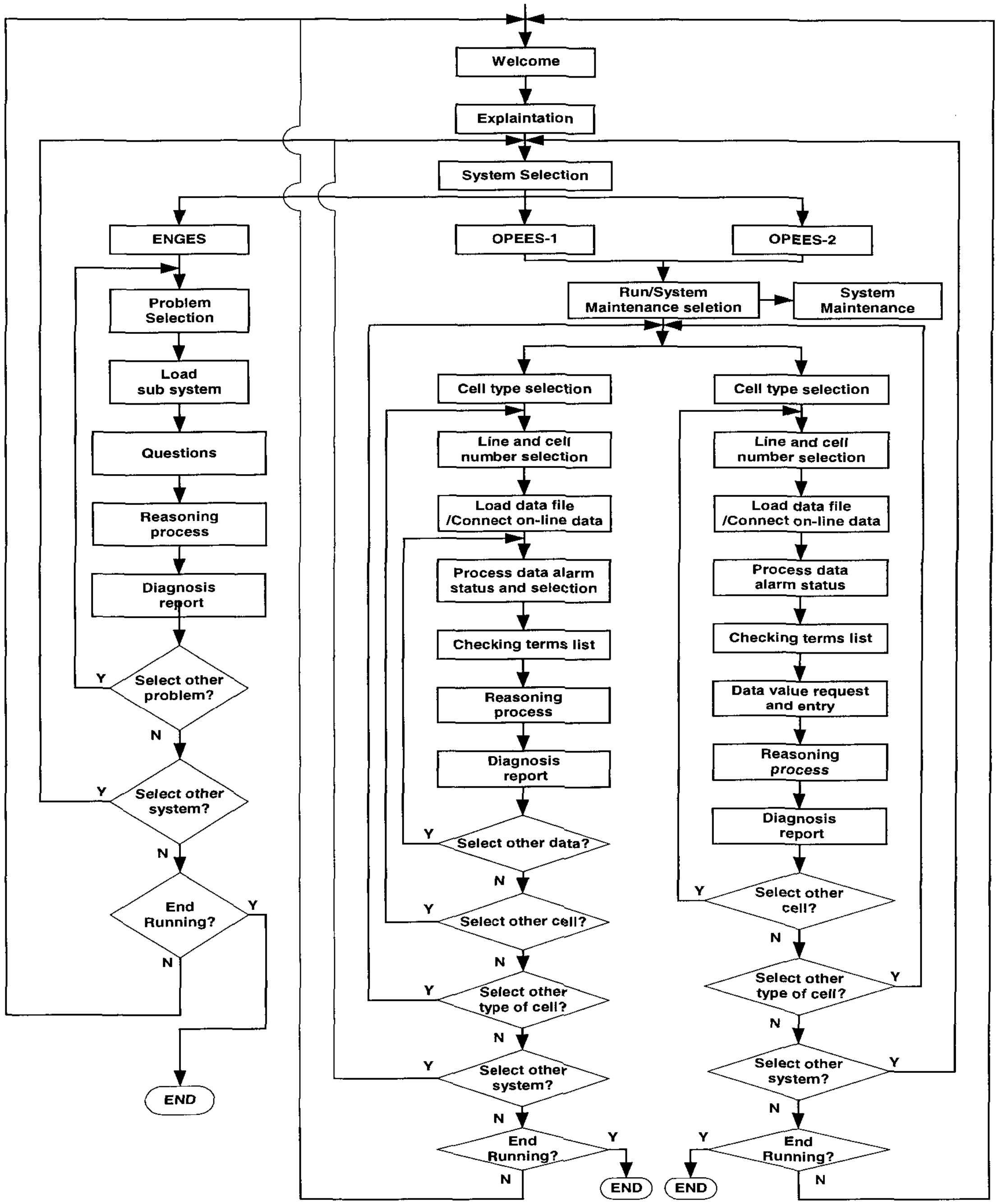

Figure A-5. 1 Flowchart of AEPES operation

APPENDIX 6 TUTORIALS OF ENGES

In this section, an example of the entire procedure of the reasoning process of ENGES (Comdale/X version) is introduced. The diagnosis problem is the "Excessive anode effect number".

When the user selects the desired problem ("Excessive anode effect number") from five available common problems, the ENGES will load the corresponding sub-system and start to run. The inference engine starts to fire the first rule in order to obtain the initial information from the user. The first question: "What is the number of anode effects during last 24 hours?" will be shown in the following question interface (See Figure A-6. 1).

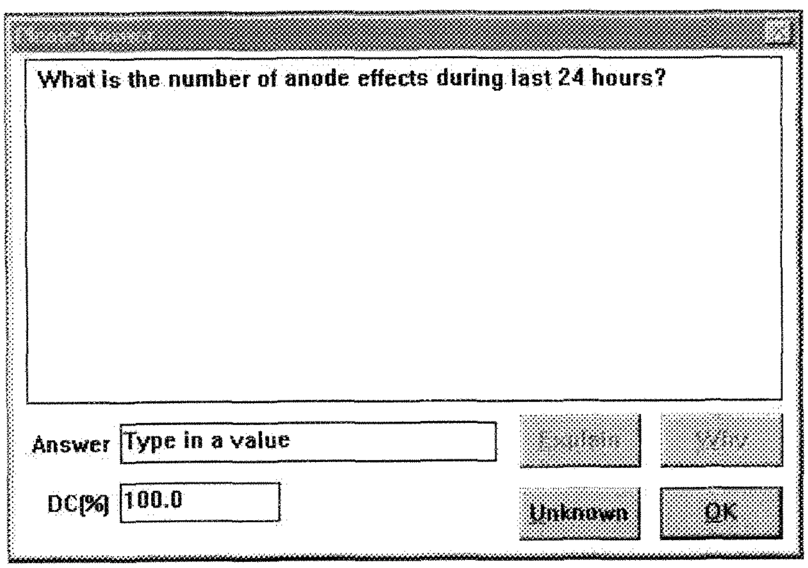

Figure A-6. 1 Question about the number of anode effects

The required value is the basic data to determine whether the cell operation is in the normal status. Depending upon the value typed by the user, the inference engine fires the rule *"anode_effect_number1"* (See Figure A-6. 2) to make the first judgment. If the number of the anode effects in the last 24 hours is more than "2", this rule would detect that the *"anode_effect.number.nomal is FALSE".*

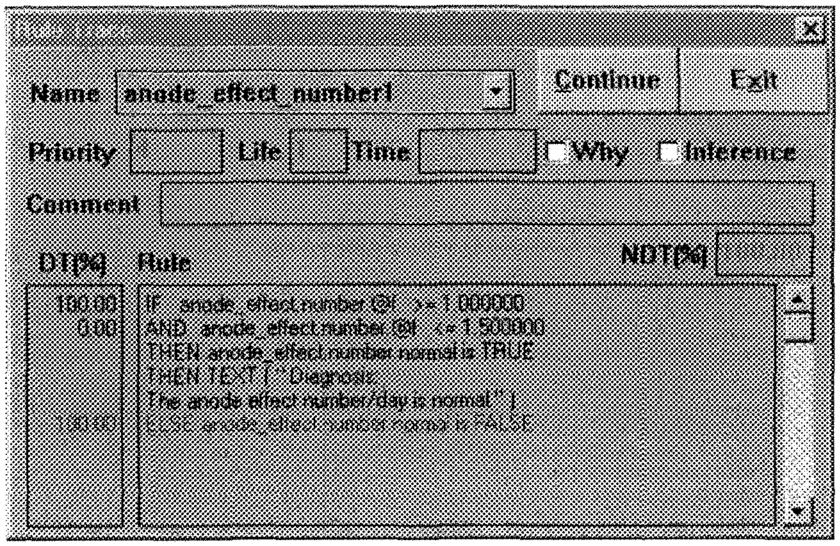

Figure A-6. 2 Example of rule trace

The first decision will then be sent into the working memory for further reasoning. The inference engine will follow the preset search strategy to find out how and when the search session is to end.

In this particular case, the forward chaining was set as the research strategy. The reasoning will restart with the first confirmed fact: "the number of anode effects is high", then it will use the relevant rules to match this fact. This means that the premises of these rules must match the first confirmed fact, as well as a chaining connection. Figure A-6. 3 illustrates the browser of the rule connection of this sub-system. The rules already executed are presented as red, while the remaining rules are still in green.

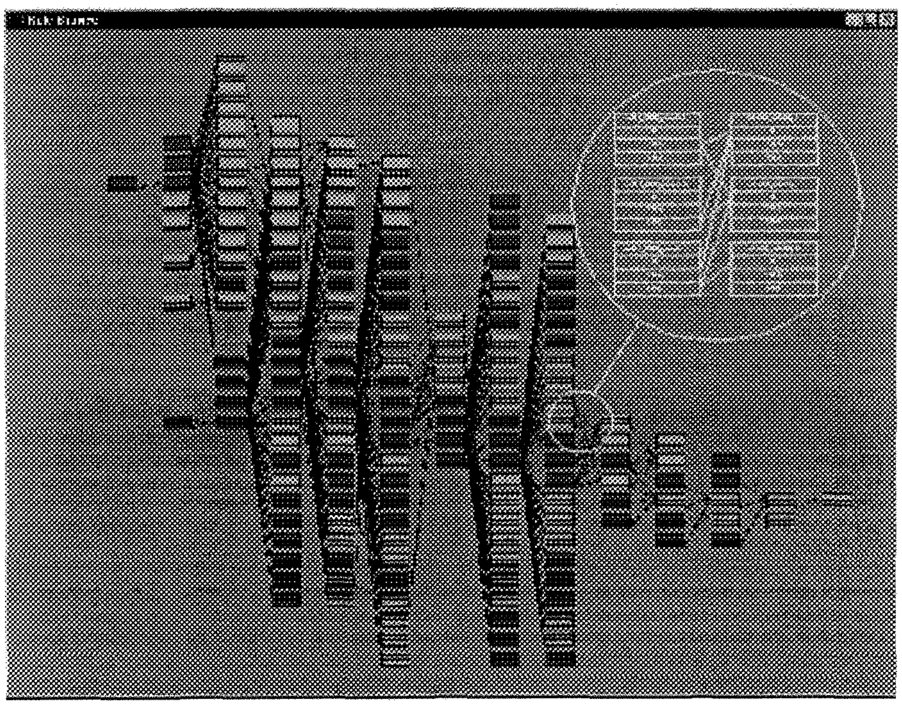

Figure A-6. 3 Example of rule browser

The inference engine will proceed from the present situation to search all the possible rules in order to find out a desired goal. While a rule is executed, the corresponding action, such as a question, conclusion, and shift command will be released one by one. For example, the Figure A-6. 4 is the consequent question released by rule *"empty_bin_feeder3."* It will check whether some obstruction occurred in the alumina bins. This question is based on the previous fact that "the number of daily anode effects is high."

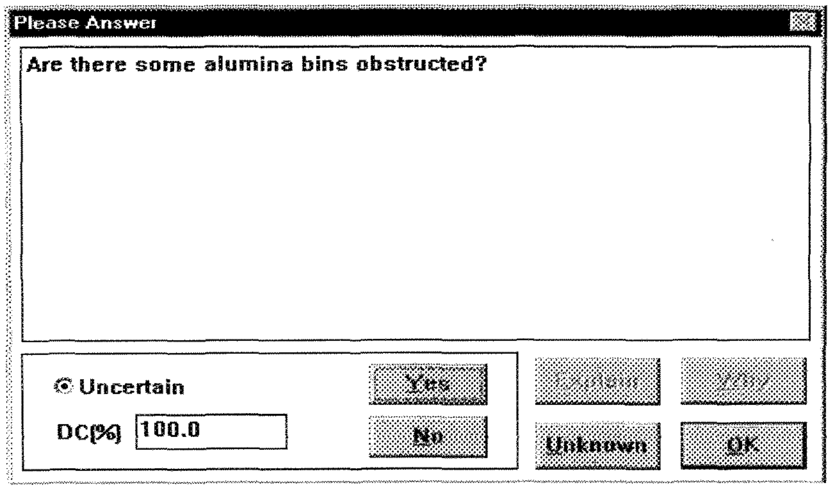

Figure A-6. 4 Example of consequent question

If some alumina bins or feeders were obstructed, the inference engine will release a corresponding illustrated suggestion (See Figure A-6. 5).

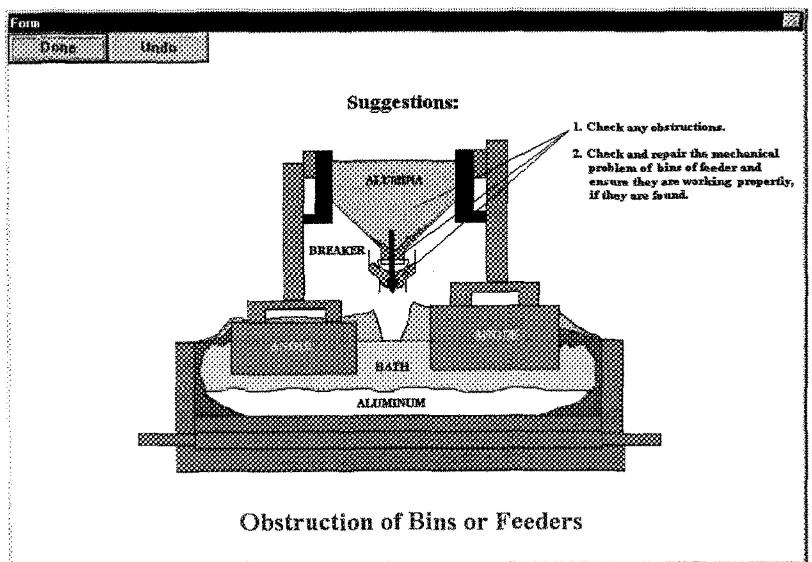

Figure A-6. 5 Suggestion about obstruction of bins or feeders

Based on these two facts, then the inference engine keeps track of the implications of exampled fragments of the rule base. The rules are organized by different priorities. For example, the rule set *"anode_effect_number"* and the rule set *"anode_effect_voltage"* are put in the high priority level, but the rule set the *"alumina_feeding_system"* is in the lower priority level. The priority of the rule set the *"alumina_property"* is in the lowest level. Figure A-6. 6 shows the different priorities of the partial rules of the rule base.

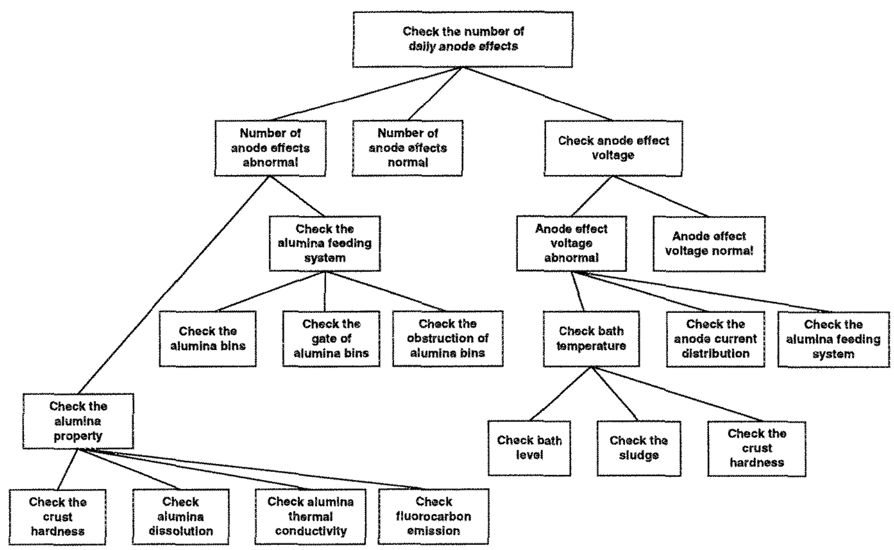

Figure A-6. 6 Different priorities of the rules

Following different priorities of the rules, the inference engine will fire corresponding rules. The following consequent questions may be selected to release depending upon the further information and the chaining link:

- Are there some alumina bins empty?
- Are there some alumina bin gates obstructed?
- Are there some alumina feeders obstructed?
- Are there some alumina feeder holes obstructed?

- Is the crust very hard and therefore more difficult to set anodes?
- Is the alumina thermal conductivity increased?
- Is the alumina dissolution decreased?
- Is the fluorocarbon emission increased?
- Is the bath level low?
- What is the value of anode effect voltage?

For example, if the *"anode_effect voltage high"* is TRUE, then the inference engine could find out an interstage diagnosis report as shown in Figure A-6. 7.

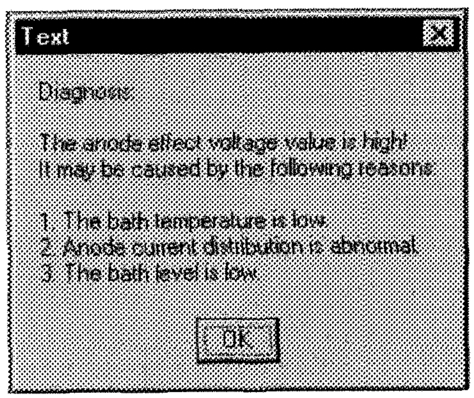

Figure A-6. 7 Example of interstage diagnosis report

In this diagnosis report, three potential reasons are listed. If the "OK" is clicked, the inference engine starts to search for the further facts to find out the underlying reasons, and then the relevant rules will be fired.

Considering these potential reasons, the following rules will be fired in the further search process:

Rule: *"anode_effect_voltage4"*
Rule: *"anode_effect_voltage5"*
Rule: *"anode_effect_voltage6"*

Each rule will ask the corresponding question to confirm relevant information. If the fact "abnormal anode current distribution" is detected, the corresponding suggestions will be released for reference. Figure A-6. 8 is the illustrated suggestion for solving the problem of "abnormal anode current distribution."

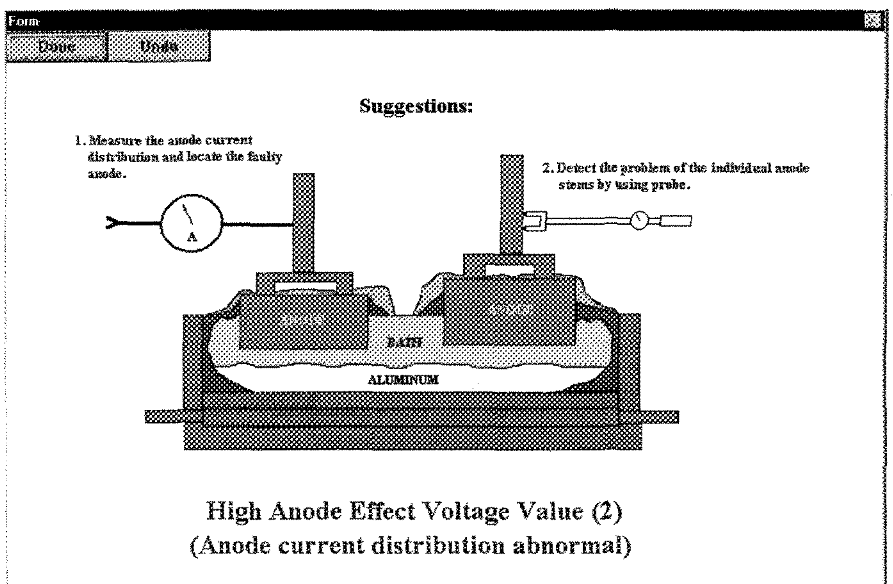

Figure A-6. 8 Suggestion for abnormal anode current distribution

When all the other possible facts have been checked, the inference engine will release a notice: "We don't find any more problems based on your answers. Terminate the diagnosis process." Then the user can click "OK" to terminate this diagnosis process (See Figure A-6. 9).

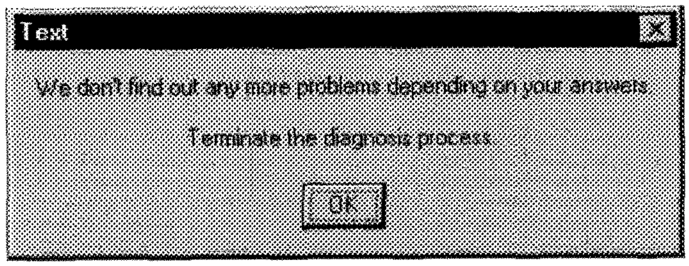

Figure A-6. 9 Termination notice

So far, this reasoning process is completed, but if the user wants continue to diagnose other problems, the user will simply follow the subsequent instruction to select the desired one and then perform as instructed.

APPENDIX 7 TUTORIALS OF OPEES

In this section two examples of OPEES are introduced. The first example comes from the basic design of OPEES and second one belongs to the advanced design.

A-7.1 Example of the basic design

Similarly to the operating procedure of ENGES, the OPEES also starts from the system selection. For this example, OPEES-1 is selected from two sub-systems of OPEES and the cell type is selected as the medium amperage, hooded prebake cell. Figure A-7. 1 shows the corresponding cell type selection interface.

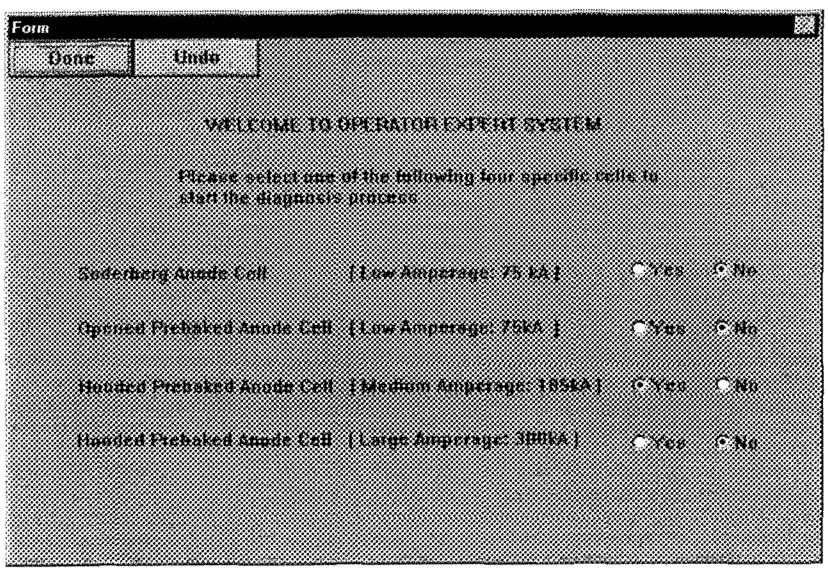

Figure A-7. 1 Cell type selection

When the cell type has been selected, the inference engine will fire a rule to load the corresponding sub-system and data file. The data file plays an important role in on-line mode of OPEES-1; all the relevant on-line data can be found in this data file, it is connected with the external process, and the stored data could be refreshed from the real process or the virtual cell.

The data file contains two types of data, the general pot line data and the process data of the designated cell. The process data is concerned with four parts: the parameter of the cell controller, the regular measurement data, the observed phenomena and the analysis data of the laboratory. Figure A-7. 2 shows an example of the data file.

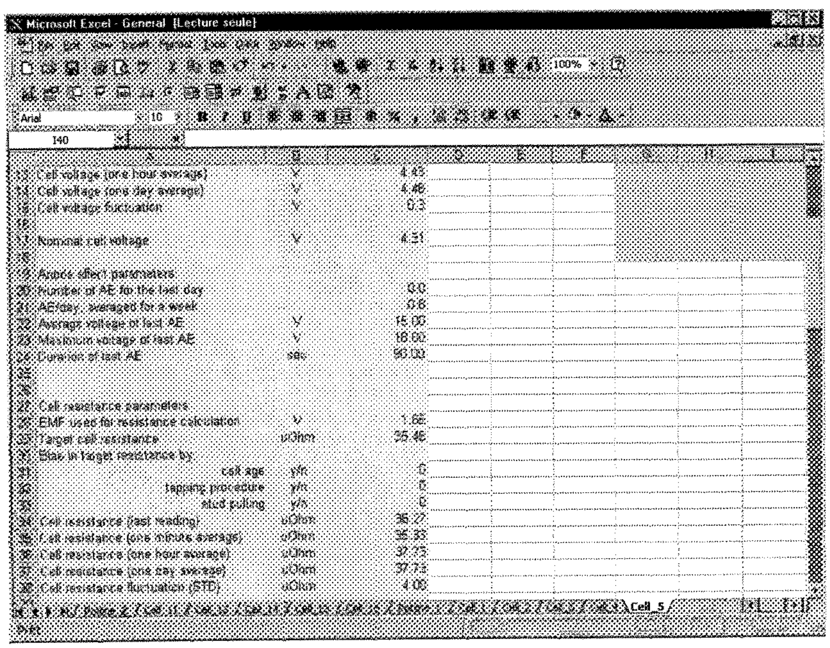

Figure A-7. 2 Example data file

The first rule executed is only for building the connection between the data source and the expert system. The alarm module of the rule base is designed to realize on-line communication. In the alarm module, there are two sub-modules: alarm module 1 and alarm module 2, which will play different roles in the process. In the alarm module 1, the rule *"alarm 1"* and *"alarm 2"* compiles all the concerned data in the data file and assigns them into the corresponding rules of the alarm module 2. Through the alarm module 2, the varied numerical value will be converted to the standard fuzzy expressions, such as "Normal," "High," "Low" and others, as well as the particular variables will be converted to the general expression. After this conversion the low level modules, which are based on the general knowledge, could be connected with the particular on-line data. (See the detailed description in the **Section 6.4.3.2 "Module hierarchy of knowledge base"**). Figure A-7. 3 illustrates this conversion process.

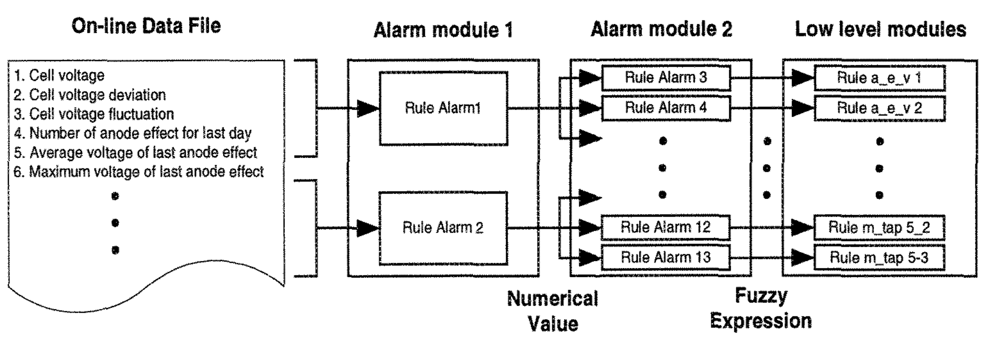

Figure A-7. 3 On-line data compiling process

Depending upon the judgments of the alarm rules, the inference engine releases a form to display the statuses of all on-line process data and requests the user to select an alarmed data to do the corresponding diagnosis. Figure A-7. 4 shows an example of the “Alarm diagnosis selection” form.

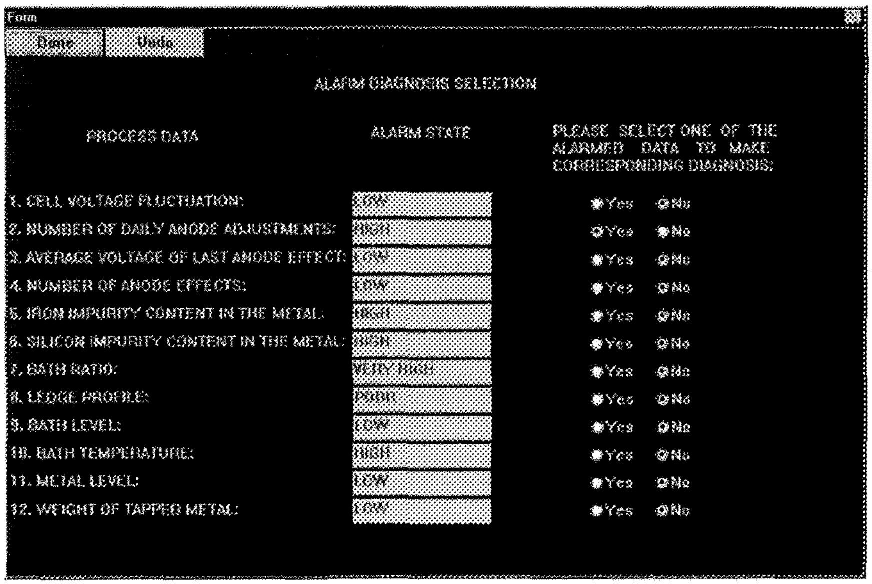

Figure A-7. 4 Example of alarm diagnosis selection

After the user selected the desired data, the inference engine will follow the search strategy to start the reasoning process. Here the backward chaining is used to fire the relevant rules. As the accessible on-line data does not include some manual measurements, in most cases, such extra information is still needed for the reasoning. For this reason, the “Check Items List” was designed to show the necessary information that cannot be obtained from the directly connected on-line data file. The required information is varied depending upon the selected alarm data. Normally, they should be taken from the potroom or laboratory. For instance, when the “Number of daily anode adjustment” has been detected as “High”, if the user wants to diagnose the cause of this alarmed data, then the corresponding "Check items list" shows in Figure A-7. 5.

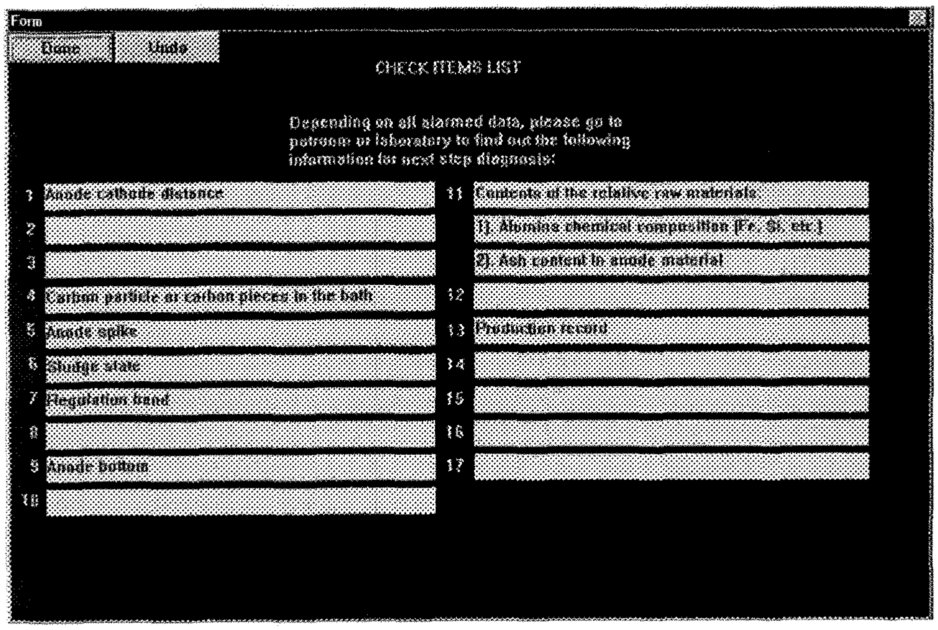

Figure A-7. 5 Example of check item list

In this list, all the required information for further diagnosis is listed at the same time to avoid the operator returning repeatedly to the potroom or laboratory. Some unquantized observations are represented by the codes. For instance, the ledge profile should be coded as in Table A-7. 1:

Table A-7. 1 Code of ledge profile

Code	Ledge profile
0	Normal
1	Poor
2	Extended

Then the inference engine will fire the consequent rules to ask questions to gather back the required additional information. Figure A-7. 6 shows an example of the question.

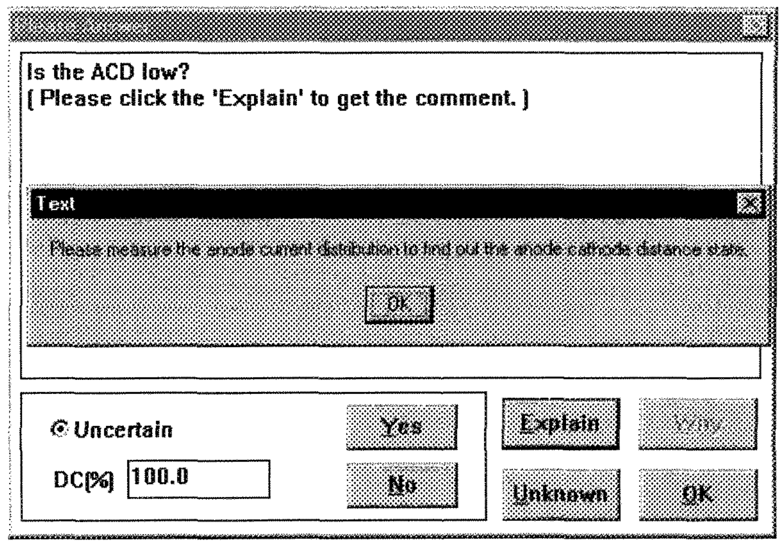

Figure A-7. 6 Example of question

ACD (anode to cathode distance) is a special parameter, which is difficult to measure in practice, therefore an “Explain” form is attached to this question interface. The user can click the “Explain” key to obtain a brief comment about how to measure the ACD; this function is also useful for the training purpose.

If the operator finds that “ACD is low,” a matched rule could be fired, depending on all relevant information and the following suggestion form (See Figure A-7. 7) released to tell the operator how to timely solve this problem. Following the suggested adjustment, the control engineer could adjust the value of set point of the cell controller. In this way OPEES-1 could work in cooperation with the existing control system. This will speed up the diagnosis procedure and improve the cell operation.

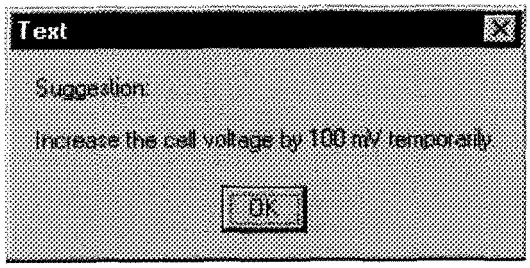

Figure A-7. 7 Example of suggestion

After this suggestion, the inference engine continues the reasoning process, and an additional suggestion, shown in Figure A-7. 8, will be released.

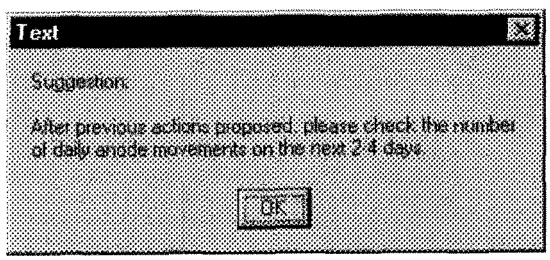

Figure A-7. 8 Example of subsequent suggestion

In some special cases, progressive suggestions will facilitate the treatment of urgent cases, as the operator needs not wait for all the suggestions until the end of reasoning process.

Because of the complexity of the aluminum electrolysis process, some potential problems may still exist after these corrections. Therefore, the diagnosis process will not stop, but will follow a similar procedure to find other possible problems. The inference engine will fire the rules to release the relevant questions. For instance, the following questions may be asked:

- Is the resistance control band abnormal?
- Is cathode voltage drop low?
- Is any component of cell voltage abnormal?

And some corresponding diagnosis and suggestions will also be released as the following indicates:

- Reset the resistance control band, if biased
- Check the anode bottom to detect the anode spike
- Remove the anode and break off spike, if present.

When all that possible problems about the required alarm data have been checked and no new problems have been found, the diagnosis process will be terminated.

A-7.2 Example of advanced design

One of the important features of the advanced design of OPEES is the improved real time communication capability. The following example shows how OPEES-2 realizes the communication through the network.

Similar to the previous example, the system starts from the sub-system selection. An example of expert system selection interface is shown in Figure A-7. 9.

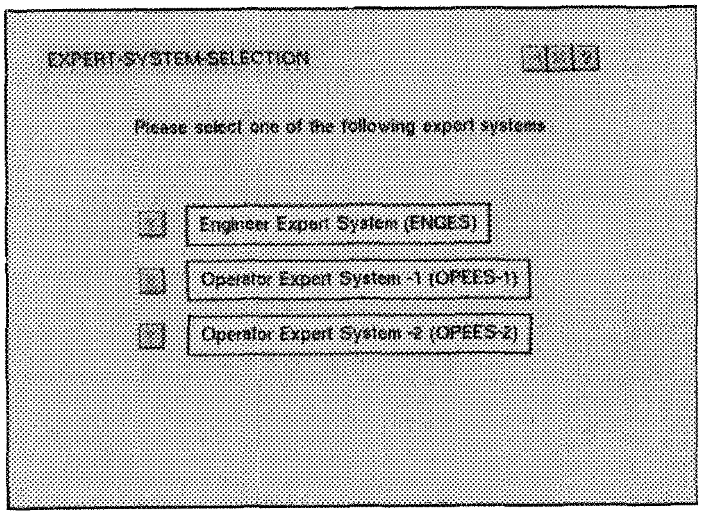

Figure A-7. 9 Example of expert system selection

When OPEES-2 is selected, the corresponding sub-system is designated. OPEES-2 is designed in accordance with the existing condition of the common smelters, where only limited number of process data can be measured automatically. For instance, in this system, the following four process data could be measured automatically and sent to the data file:

1. Cell voltage fluctuation
2. Number of daily anode adjustments
3. Average voltage of last anode effect
4. Number of anode effects last day.

The basic design takes advantage of Microsoft Windows applications, keeping the same type of data files, which were used for the basic design. To improve the interaction with other applications in a network environment, the G2 ActiveXLink is chosen as the communication bridge, which can integrate the G2 and the Microsoft Windows formatted data files through the network.

ActiveXLink in the knowledge base of OPEES-2 provides a facility that allows browsing the procedures defined in the G2 knowledge base through the network, which means, the user in the smelter can implement the interactive communication with OPEES-2 through the network (See Figure A-7. 10).

Therefore, using ActiveXLink accelerates the delivery of solutions in the Windows environment, which also benefits the corporation with other users in the network.

When the line and cell number are selected, the inference engine will invoke the corresponding procedure to integrate the OPEES-2 and the Excel data file, which is located in the site of virtual cell or real smelter, as well as to set up a communication bridge. The invoked procedure *"Get-process-data-value"* (See **Section 7.3.2 "On-line diagnosis"**) will send a message to the user at the cell to get the process data. This message could be displayed on the Excel spreadsheet there (See Figure A-7. 10).

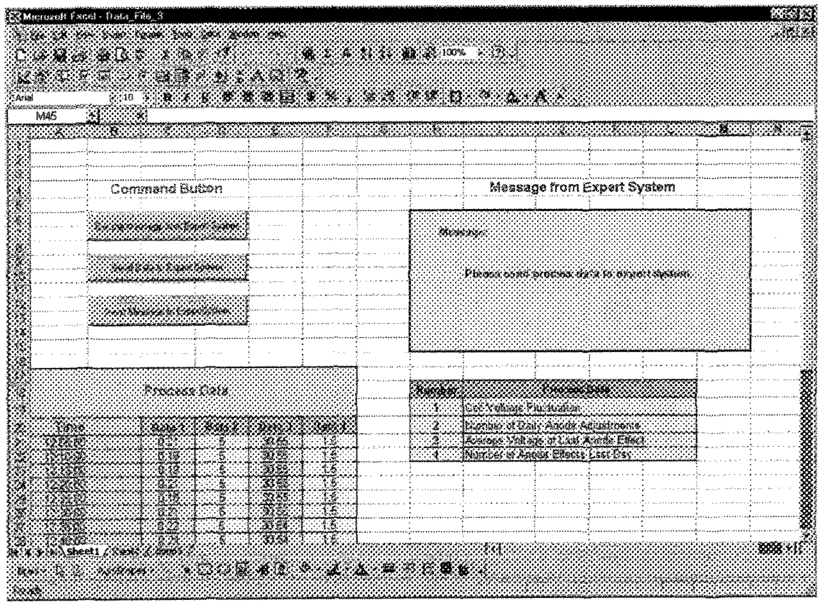

Figure A-7. 10 Example of data file

When the operator presses the button "Send data to expert system," a group of on-line process data will be sent to the OPEES-2 through the network. These data also can be automatically refreshed at regular intervals.

The inference engine will fire the alarm rules to judge the status of connected data, and release an "Alarm Status List" form to show the values and operating status of these received process data (See an example in Figure A-7. 11).

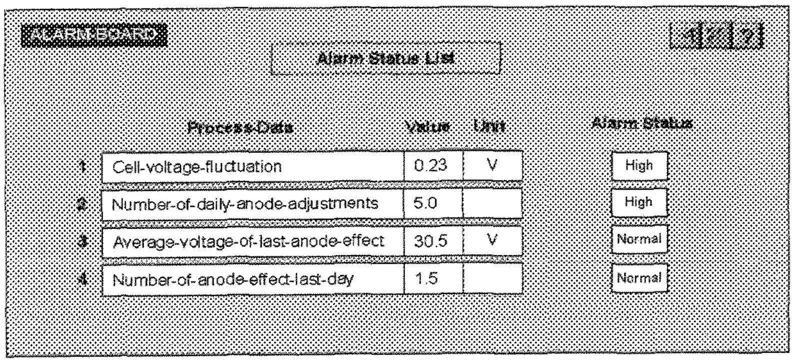

Figure A-7. 11 Example of on-line data alarm status

Considering the fact, that only limited on-line data is available, and that there is not enough information to do the reasoning for that moment, there is no diagnosis data selection form in OPEES-2. The entire diagnosis will not start until all necessary information has been gathered. To check and collect the required information, a "Check Items List" form will be released for reference, which lists the required measurements in the potroom and the data analyzed in the laboratory (See Figure A-7. 12).

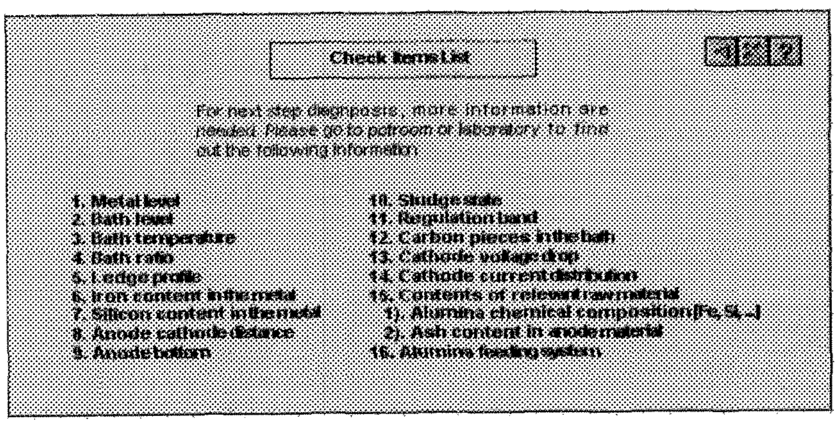

Figure A-7. 12 Check Items List

Following this list, the operator should go to the potroom and the laboratory to make the corresponding measurements and get the analysis results. When all the required information is obtained, the operator can start to answer the questions one by one. These questions are related to the status of on-line and manually collected data. Figure A-7. 13 is an example of the questions.

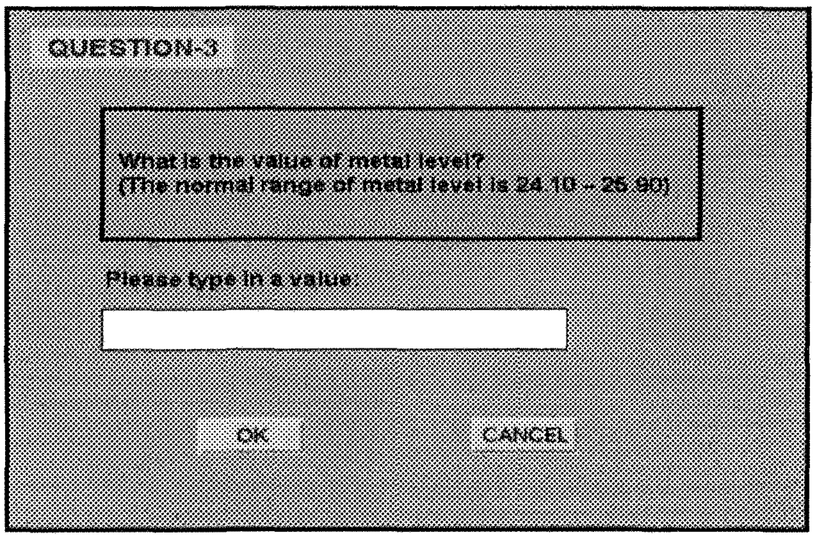

Figure A-7. 13 Example of question of OPEES-2

This question differs from the previous one. A numerical value is requested here.

The reasoning procedure is similar to OPEES-1. When the problem is found, the diagnosis report and operating suggestions are given and the messages can be sent to the operator in the potroom. Figure A-7. 14 is an example of the suggestions, in which detailed operations are listed to tell the operator how to solve the problem step by step.

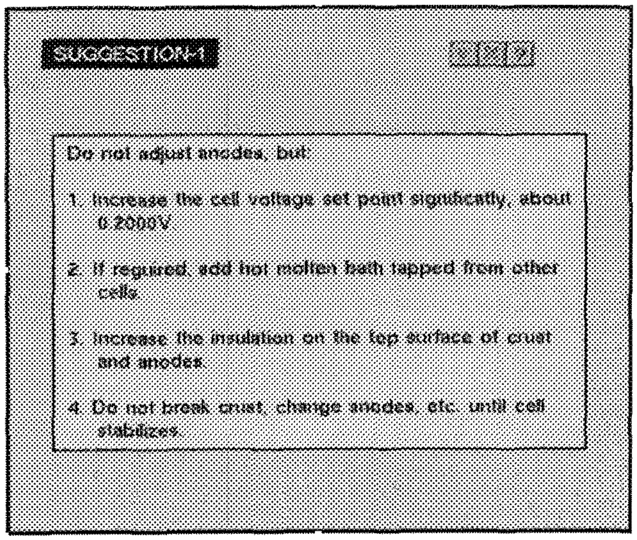

Figure A-7. 14 Example of suggestion

If no more problems are found depending upon the available information, the diagnosis process will terminate.

A-7.3 Discussion

The description of these two examples shows how the system would run under an ideal condition, but for the practical application, more complex cases could be encountered. To ensure the diagnosis process is successful, the following aspects are considered as the restrictive conditions:

- All the requested information should be provided
- Answers, especially numerical values, should be within reasonable range

- The system will give unreasonable results if many questions are not answered correctly

That means even though the system has been designed with some tolerance for the uncertainty and unclear information, but if they exceed the limitation of tolerance that could still lead to a fault in the diagnosis process.