TEMPORAL REASONING IN REAL-TIME EXPERT SYSTEMS

by

Karen L. Stephens

TR91-063 April 1991

This is an unaltered version of the author's
M.Sc. (CS) Thesis

Faculty of Computer Science
University of New Brunswick
P.O. Box 4400
Fredericton, N.B. E3B 5A3

Phone: (506) 453-4566
Fax: (506) 453-3566

TEMPORAL REASONING IN REAL-TIME EXPERT SYSTEMS

by

Karen L. Stephens

BSc(CS), University of New Brunswick, Canada, 1983

A THESIS SUBMITTED IN PARTIAL FULFILMENT OF

THE REQUIREMENTS FOR THE DEGREE OF

Master of Science in Computer Science

in the Faculty

of

Computer Science

This thesis is accepted.

Dean of Graduate Studies and Research

THE UNIVERSITY OF NEW BRUNSWICK

April 1991

ACKNOWLEDGEMENTS

I wish to express my gratitude to my thesis supervisor, Dr. B. Nickerson for his guidance and much needed encouragement during the course of the conceptualization, design and implementation of this thesis.

I also acknowledge the help and support provided by John E. Smith of AECL Candu, Maritime Nuclear. As well, I wish to acknowledge the staff at the Dalhousie Generating Station, especially Bob Quondum who as my liaison at the plant helped to provide much of the technical information required to successfully complete this thesis.

I would like to express a special thanks to my friends and colleagues for the help and encouragement received. Finally, I would like to thank my family for the willing sacrifices they have made to see this thesis to its successful completion.

DEDICATION

To

my husband, Gary

and

my daughters, Megan and Amanda

ABSTRACT

This thesis presents research and development of a temporal logic for use within real-time expert systems. The temporal logic provides a framework to represent both past and present temporal information. A prototype expert system was built to demonstrate the feasibility of these ideas in the domain of sootblowing within a coal-fired power generating station.

A real-time expert system prototype has been developed to advise operators of sootblowing strategies by monitoring the performance of the boiler and the build-up of slag. This prototype called the Sootblowing Advisory Expert (SAX) includes several components. The first component is the Temporal Preprocessor which generates temporal events and relationships from operator entered information and simulated process I/O. The second component, Temporal History, maintains the history of temporal events and relationships. The Query Processor provides the means to transparently query both the history and current temporal history databases. The fourth component of SAX implements the domain specific dynamic and static expert knowledge in the form of facts and rules. The last element of SAX is the User Interface which provides the operator with the means to enter interactively plant operating information unavailable through process I/O. As well, the user interface provides the sootblowing advice and status information.

This system was developed using the ART-IM Automated Reasoning Tool developed by Inference Corporation and is functional on a PS2/Model 70.

CONTENTS

LIST OF FIGURES

LIST OF TABLES

NOMENCLATURE

AECL	Atomic Energy of Canada Limited
RTES	Real-Time Expert System
SAX	Sootblowing Advisory Expert
UI	User Interface
Prim	Primary Sootblowing Strategy
Frhtr	Final Reheat Sootblowing Strategy
Hrhtr	Hot Reheat Sootblowing Strategy
Crhtr	Cold Reheat Sootblowing Strategy
Sshtr	Secondary Superheater Sootblowing Strategy
Uww	Upper Waterwall Sootblowing Strategy
Mww	Middle Waterwall Sootblowing Strategy
Lww	Lower Waterwall Sootblowing Strategy
PS_1	Plant State 1
PS_2	Plant State 2
PS_3	Plant State 3
PS_4	Plant State 4
PS_5	Plant State 5
PS_6	Plant State 6
PS_7	Plant State 7
PS_8	Plant State 8
PS_9	Plant State 9
PS_10	Plant State 10
PS_11	Plant State 11
PS_12	Plant State 12
PS_13	Plant State 13
PS_14	Plant State 14
PS_15	Plant State 15
PS_16	Plant State 16
PS_17	Plant State 17
PS_18	Plant State 18
PS_19	Plant State 19
PS_20	Plant State 20

1 INTRODUCTION

1.1 Real-Time Systems

Real-time computing systems are being used in many industrial, military and space applications. A real-time system is one in which the correctness of the system depends not only on the logical result but on the time at which the result was produced. The design of upcoming real-time systems will include many of the following features [STAN88]:

1. Distributed
2. Highly dynamic and adaptive
3. Exhibit intelligent behaviour
4. Failure to meet timing requirements will be considered catastrophic

It is sometimes useful to categorize real-time systems into hard and soft real-time. This classification provides a means of distinguishing the basic characteristics of the real-time system in question. As described above, a hard real-time system which operates correctly is one which depends both on the result and the time at which the result was produced. Failure to meet requirements of a hard real-time system is potentially disastrous. A soft real-time system operates under timing constraints for which failure to meet the constraints is not dangerous. In fact, it may be useful to complete the current task even though its time has been exceeded [LAFF88].

1.2 Real-Time Expert Systems

An expert system is a computer program that emulates the behaviour of a human expert in a specific domain of knowledge. Knowledge in any speciality can be classified as public or private. Public knowledge includes published definitions, facts, procedural rules and theories. Human experts generally possess private knowledge which has not found its way into the published literature. This knowledge generally consists of rules of thumb which have come to be known as heuristics. The main characteristics of expert systems are identified as follows:

1. Extensive specific knowledge from the domain of interest
2. Application of search techniques
3. Support for heuristic analysis
4. Symbolic Processing
5. Explanation of its own reasoning

Expert system technology requires the separation of the knowledge from the mechanism which processes the knowledge. The knowledge base contains all the knowledge which is required for decision making. The inference engine is an algorithm which drives the thinking process and organizes the order in which knowledge is considered to solve any problem. Generally associated with these two components is a user interface mechanism which allows a user to communicate with the expert system.

To address the complex problems which are currently being encountered in real-time system development, expert system techniques are being explored as a means of providing more sophisticated strategies. One of the characteristics of complex real-time systems is the vast amount of data that must be assimilated before decisions can be made. These decisions must be reached within the timing constraints of the application. This places a large cognitive load on the individuals involved. Cognitive overloading has been targeted as one of the major reasons for the use of expert systems in real-time applications.

A real-time expert system needs to respond to a changing task environment involving an asynchronous flow of events and dynamically changing requirements with limitations on time, hardware and other resources. There are many problems which require consideration when designing a real-time expert system [LAFF88]:

1. Nonmonoticity - ability to handle program data changes during program execution as new events occur.

2. Continuous Operation - systems must operate until stopped by an operator or a catastrophic event occurs.

3. Asynchronous events - ability to prioritize events and process asynchronous events.

4. Interface to an External Environment - ability to interface to sensors.

5. Uncertain or Missing Data - capability of processing data when information is incomplete as well as when the validity of the data has diminished.

6. High Performance (Speed) - ability to respond to rapidly changing data.

7. Temporal Reasoning - ability to reason about past, present and future events as well as their sequence.

8. Focus of Attention - ability to focus the system on the data of greatest importance.

9. Guaranteed Response Time - respond to events by the time the response is needed.

10. Integration with Procedural Components - typically interface to procedural software to perform some of the functions of the real-time system to which knowledge based approaches are not suited.

1.3 Temporal Reasoning

The passage of time leads to changes. Representation of temporal information and reasoning about such information, requires a method which will capture the concept of change over time and can express the truth or falsity of statements at different times [SHOH88].

Reasoning about time can be found in many areas of artificial intelligence. Prediction of the future, explanations of why current situations exist, planning of events to meet a specific goal and learning rules based on a time sequence of events are some of the areas in which reasoning about time is prevalent.

1.3.1 **Change-Based Approaches to Temporal Reasoning**

There are two main methods for expressing change over time. They are the change-based and time-based approaches.

Situational calculus, introduced by McCarthy and Hayes in 1969, maintains a set of states where each state is achieved when an event occurs that causes the current state to be modified. The example of removing slag from the boiler walls illustrates the state concept. *Slag on the lower west waterwall* is state A. Sootblowers X,Y and Z are operated which is an event which will modify the current state. State B, *No slag on the lower west waterwall*, is then entered.

Pratt introduced dynamic logic which is a representation for reasoning about events based on modal logic. The basic idea of dynamic logic is to represent programs as modal operators which when applied to a program state will change it to a new program state. These programs are change indicators. Modal logic can represent past, present and future states and requires a large number of changes when an event occurs.

1.3.2 **Limitations of Change-Based Approaches in Temporal Reasoning**

Situational calculus as well as other change-based approaches suffer from the limitation of expressing time only in the sequencing of change indicators (ie. moving from state to state). Limitations also exist in the ability to express many properties of actions or events. All actions are instantaneous and have no duration. All effects of an action are immediate and eliminate the expression of delayed effects or natural death which is an effect which automatically stops without the requirement of further action. Concurrent actions cannot be represented. As well, the ability to express continuous processes is not possible since time is discrete and is specified as time points.

Other problems which occur are more general in that irrespective of the formalism used to express time these problems exist and must be addressed. The first of these problems is qualification which involves predicting the future without taking into account everything in the past. The second issue which arises is called ramification and deals with actions which are complex and whose effects are difficult to define completely. Lastly, the frame problem occurs as a result of trying to express all the things which did not change as a result of an action.

1.3.3 Time-Based Approaches to Temporal Reasoning

In time-based approaches the passage of time is the only element of change. One can make an assertion which can be either true or false at a point in time. To develop a temporal logic which formalizes the passage of time we must first define the characteristics of time and then the formalism in which the temporal information will be expressed. For instance, assertions can be interpreted at a point or over an interval. The structure of time can be, for example, linear or circular. Some of the current formalisms used to express temporal information are classical first order logic [MCDE82] [MCDE80] [ALLE84], reified sentences [SHOH88] and modal temporal logics.

Allen

Allen relates his temporal logic to the meaning of English sentences that describe actions and events. This can be used for action reasoning (problem solving) [ALLE84]. He views the world as a set of temporally qualified assertions. These assertions outline what is known about the past, present and future. The temporal logic is formalized as a first order predicate calculus. The most important predicate is *HOLDS(p,t)* which expresses that *a property p holds over the time interval t.* Time is linear and assertions are interpreted over time intervals. Information in the world is either static, expressed as properties, or dynamic, expressed as occurrences allowing for change over time. The basic primitives of the logic express the relations which can occur between time intervals. An example of this is *DURING(t1,t2)* which translates to *the time interval t1 is fully contained in t2.* Based on these primitives, axioms have been defined expressing their behaviour.

Occurrences are objects in Allen's logic which can be divided into processes and events. A process is an occurrence in which no result is anticipated and is expressed with the *OCCURRING(p,I)* predicate. An example is *OCCURRING(slag_building_up,I)* which translates to *slag is building up in the boiler over the interval I.* An event is an occurrence which has an outcome or product expressed as *OCCUR (e, I).* An example is *OCCUR(falls(slag, waterwall, slope), I)* which denotes that *slag has fallen from the waterwalls to the slopes over the interval I.*

McDermott

McDermott also presents a first order temporal logic in which the two key ideas are the openness of the future and the continuity of time [MCDE82]. His research attempts to express causality, continuous change and planned actions. Time is infinite and noncircular. Assertions or states as McDermott calls them are point based. The universe is an infinite collection of states where a state is an instantaneous snapshot of the universe. Associated with each state is a time of occurrence. Axioms are defined which relate the ordering of states one to the other and the properties which hold. States are organized into chronicles where a chronicle is a totally ordered set of states. There are many different chronicles giving rise to a vast number of possible futures.

Objects in McDermott's logic which operate on states and chronicles are facts and events. A fact is a set of states (states in which the fact is true) which can change in truth value over time. *Fact p is true in state s is denoted by (T s p).* An event causes a change in a fact and is a set of intervals over which the event happens. An interval is a totally ordered convex set of states. *Event e happens between states S1 and S2* is expressed as *(Occ S1 S2 e).*

Causality is defined to mean that one event type follows another event type denoted *(Ecause p e1 e2 rf i).* This is interpreted as *e1 always follows e2 after a delay in the interval i unless p becomes false before the delay is up.* Delay is measured from rf through e1.

Continuous change is represented by fluents. The fundamental event involving fluents is a vtrans expressed as *(vtrans v r1 r2)* meaning the *event consisting of all occasions when v changes from r1 to r2.* Planning deals with the world of actions denoted by McDermott as *(do agent act)* which is *an event consisting of an agent performing an action.*

<u>Shoham and Goyal</u>

Shoham and Goyal developed a reified first order temporal logic which expresses temporal propositions over time intervals. The basic temporal component is a point

[SHOH88]. An interval is defined by its end points. An instance in time is an interval with zero duration. Time is linear in nature. The basic form of a temporal proposition is *True(t1, t2, p)* which is the *proposition p is true over the interval bounded by t1 and t2*. Shoham and Goyal present a syntax and semantics for their temporal propositions.

Propositions are divided into two types, event-like and fact-like propositions. Distinction of propositions types is specified by how the truth of a proposition over an interval is related to the truth over other intervals. Rules governing the truth of propositions are specified by several characteristics. Downward-heredity is an example characteristic which states whenever a proposition holds over an interval it holds over all its sub-intervals.

Trudel

Trudel's research resulted in the development of the first order temporal logic GCH [TRUD89a] [TRUD89b] [TRUD89c] [TRUD90]. In this logic the basic representation of time is a point. The underlying temporal point structure is continuous and thus was chosen as the real domain.

The truth of an assertion must be reasoned about at a point in time or over an interval. A point assertion is expressed via the true relation. An example of the true relation is *true(t1, on(slag, boiler waterwall))* which denotes that *the relation slag on the boiler waterwall is true at point t1.*

The truth of an assertion over an interval is directly related to the truth of the assertion over all the included sub-intervals. An interval based assertion is given by the Reimann Integral. An illustration of an interval based assertion is *integral(t1, t2, on(slag, boiler waterwall), x)* which means that *the variable x is the amount of time the relation, slag on the boiler waterwall is true between the points t1 and t2.*

Trudel illustrates the usefulness and completeness of GCH in the areas of persistence, interval representation and within the qualitative and quantitative domains.

Balasubramanyan

Database management is also an area in which the representation of time has become prevalent. Balasubramanyan presents a historical relational database management system which supports the representation of the temporal dimension for real world data [BALA89]. In the database model time is a linearly order set modeled after real numbers for continuity. Objects in the historical database are represented as relations where timestamps are attributes of the relation. The Historical Relational Data Model is characterized by:

```
1.    State of a database object
2.    Real world time modelling
3.    State and event database objects
4.    Interval stamping of tuples
5.    Partitioned database
6.    Historical query language
```

The basic facilities offered by the HRDBMS are time modelling, historical relational algebra and historical structured query language.

Kowalski

Kowalski has also performed research in the area of database updates and narrative understanding [KOWL85] [KOWL86]. He outlines an approach for reasoning about events and time within a logic programming framework intended for maintaining and updating databases. He formalizes time using event calculus which is expressed in Horn clauses, a subset of classical logic augmented by negation by failure. Event calculus has the following general characteristics:

1. Updates to the database simply add information regarding the relationships. Updates are not destructive.
2. Derivation of starts and ends of relationships from event descriptors imposes an extra level of semantic structure on database updates.
3. Past and future are treated symmetrically and thus event descriptors can be assimilated in any order.
4. Events are ordered relative to each other, that is E1 < E2 means that E1 occurred before E2.
5. Events, time periods and time instances can be defined which ultimately lead to the ability to express concurrent events.
6. Events can have duration. Events occur after the periods they end and before the periods they start and are not fully contained within them.
7. The frame problem is addressed by qualifying relationships with time periods instead of global situations (situational calculus).
8. In event calculus time periods are the entity over which assertions are made.

The basic building blocks in the event calculus are time periods, relationships and events. Associated with a time interval is its relationship. For example *R(X, Y, after(E1))* means a *relation R operates on attributes X and Y during the time period after (E1)*. Time periods are started or ended by events. As an example, after(E1) is a time period expressed in terms of event E1. This method will be discussed further in Chapter 2.

1.4 Steam Temperature Control

The problem domain chosen for research in this thesis is Steam Temperature Control in coal-fired generating units, and in particular, sootblowing which controls the buildup of slag on the heat transfer surfaces of the boiler. Accurate regulation of steam temperature in coal-fired generating units will improve the heat rate of the boiler due to the high cycle efficiency possible with high steam temperatures. It will also prevent failure of various parts of the boiler (eg. superheaters, reheaters or turbine), prevent thermal expansion of the turbine and avoid erosion from excess moisture in the last stages of the turbine.

Maritime Nuclear, a branch office of AECL CANDU, is currently developing a prototype real-time expert system for Improved Steam Temperature Control in coal-fired generating units [MARI90]. The improved steam temperature control system will be tested under simulated conditions. The simulation itself will be provided by a **process model** of a coal fired unit. With the cooperation of NB Power, Unit #2 at the Dalhousie Plant will be used as the target coal-fired unit for the process model. Unit #2 is a Combustion Engineering corner fired reheat unit of 215 Megawatts net, designed for both oil and pulverized coal firing. The burners are of tilting tangential type arranged on four levels. Each level is supplied by a separate pulverizer.

1.4.1 Sootblowing for Improved Steam Temperature Control

One of the key areas where an improvement in steam temperature control can be realized is the control of slagging on the heat transfer surfaces of the boiler. When coal is burned it produces slag as a by-product. Due to the high temperatures generally reached in most boilers this slag becomes quite sticky. Slagging is the build-up of this sticky material on the surfaces of the boiler. As a result of slagging, the efficient transfer of heat to the water side of the boiler is inhibited. The ultimate implication of this phenomena is the control of steam temperature becomes very difficult.

Slag build-up in a boiler can be attributed to several factors found in typical coal fired units:

```
1.  Burner Tilt
2.  Coal Quality
3.  Mills in Operation
4.  Excess Air
5.  Fuel Flow
```

1.4.2 Sootblowing Operation

Sootblowing is the method by which slagging is removed from the heat transfer surfaces of the boiler. This is not an automatic control mechanism. Sootblowing is manually initiated by an operator when he/she has determined that slag has accumulated and is affecting the efficient operation of the boiler.

Sootblowing itself is the release of steam jets directed to specified areas of the boiler. This will cause the slag in that area of the boiler to break and fall away.. Approximately 90 sootblowers are distributed throughout the waterwalls, superheat and reheat sections of the boiler as indicated in Figure 1-1.

1.5 Scope of Thesis

This thesis deals mainly with the implementation of a temporal logic to provide a means of performing temporal reasoning in real-time domains. Towards this end a prototype real-time expert system called the Sootblowing Advisory Expert (SAX), using the developed temporal logic, was designed and implemented to monitor and control slag buildup in a coal-fired power generating station.

The material in this thesis is presented in the following manner. Chapter 2 presents the background and theory of the temporal logic implemented in SAX. Chapter 3 then discusses the implementation of this temporal logic. Maintenance, design and implementation of temporal history is discussed in Chapter 4. The design and implementation of SAX is presented in Chapter 5. Chapter 6 presents the testing procedures, the tests performed, test results and provides an analysis of the performance of SAX. Chapter 7 completes this thesis with concluding remarks and discusses possible future work in this area.

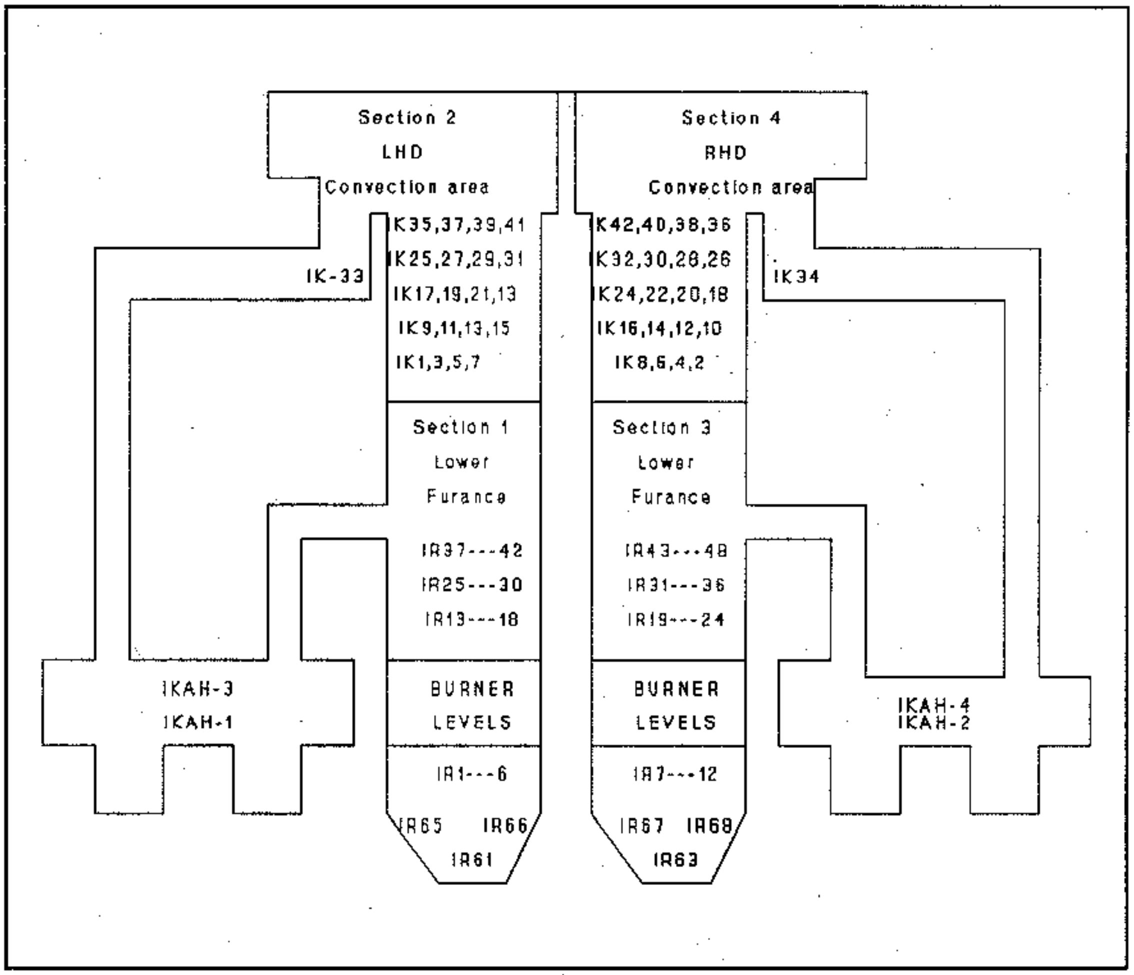


Figure 1-1 Distribution of Sootblowers in Boiler

2 EVENT CALCULUS FOR REAL-TIME TEMPORAL REASONING

2.1 An Overview Of Event Calculus

Event calculus was developed by Kowalski and Sergot [KOWA85] [KOWA86] for the purpose of expressing database updates and narrative understanding. It is:

```
1.   Based on the treatment of time

2.   Based on the notion of an event

3.   Formalized in HORN clauses which are a subset of
     classical  logic.  This  is   supplemented  with
     negation as failure.
```

The basic elements of event calculus are time periods, events and relationships. These elements are interrelated in that an event initiates and terminates one or more time periods. Associated with a time period is a relationship. A time period is expressed as a function of the event in which it is initiated or terminated. For example,

```
An  event  e  of  operating  sootblower  x  starts  a
period of time for  which x has availability y.
```

Refined as a HORN clause, this is represented as:

```
x has availability y for period after( e ) if e is
an event in which x is operated.
```

The time period after(e) is initiated by event e and is expressed as *start(after(e) e)*. The termination of a time period after(e) is an event e' and is denoted as *end(after(e) e')*. Associated with an event is a time of occurrence represented by *time(e January 12,1991)*. Figure 2-1 is a pictorial representation of the time period after(e).

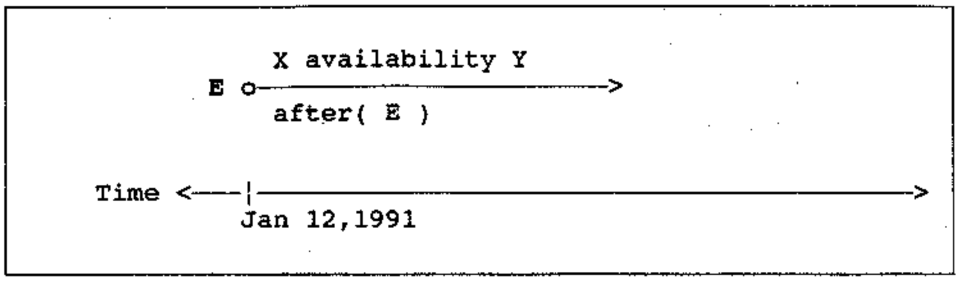


Figure 2-1 Time Period Representation

Each time period is symmetrically defined. For every after(e) time period there is a corresponding before(e') time period where e' is the event that terminates after(e) and e is the event that terminates before (e'). This is illustrated in Figure 2-2.

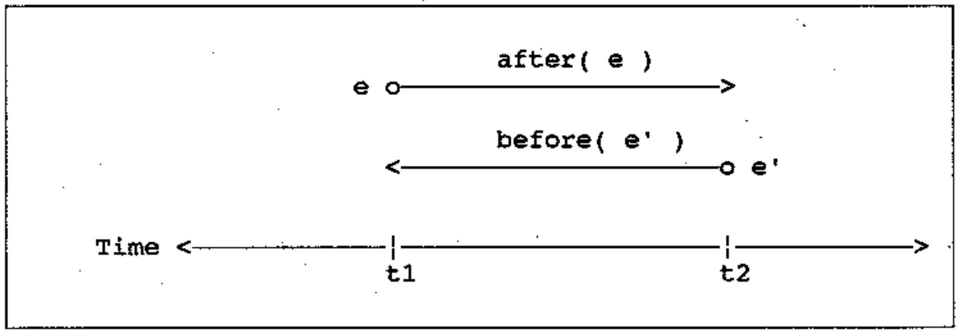


Figure 2-2 Symmetrical Time Periods

The start and end of time periods is governed by the type of event and the relationships associated with the time period. A time period persists as long as an event does not occur which terminates the relationship associated with the time period.

2.2 An Example of Event Calculus

In this section an example will be presented which will help illustrate the basic concepts of event calculus, in particular, events, relationships and the initiation and termination of time periods.

In this example the relationship *FuelBurn* will be manipulated. This relationship expresses the type of fuel being burned at a coal-fired generating station. The following rules govern the initiation and termination of time periods for the relationship *FuelBurn*.

1. An event e of a shutdown in a plant x terminates a time period where fuel y was being burned in plant x and initiates a time period where no fuel is being burned in plant x.

   FuelBurn(x y before(e)) if PlantShutDown(x y e)
   FuelBurn(x NONE after(e)) if PlantShutDown(x y e)

2. An event e of bringing a plant x online initiates a time period for which y is the fuel being burned in plant x.

   FuelBurn(x y after(e)) if PlantStartup (x y e)

3. An event e of switching from fuel y to fuel z in plant x ends a period of time for which fuel y is burned in plant x and starts a period of time for which fuel z is burned in plant x.

   FuelBurn(x y before(e)) if SwitchFuel (x y z e)
   FuelBurn(x z after(e)) if SwitchFuel (x y z e)

The following events occur and are shown as successive updates in Figures 2-3 to 2-7.

1. E1 is an event in which Dalhousie Unit #2 is brought online with Minto Coal as the fuel. E1 occurred Nov 10, 1989.

   PlantStartup(DalhousieUnit#2 MintoCoal E1)
   time(E1 Nov10,1989)

2. E2 is an event in which Dalhousie Unit #2 is shutdown and the burning of Offshore Coal is terminated. E2 occurred May 10, 1990.

   PlantShutDown(DalhousieUnit#2 OffshoreCoal E2)
   time(E2 May10,1990)

3. E3 is an event in which the fuel in Dalhousie Unit #2 was switched from Minto Coal to Offshore Coal. E3 occurred Jan 13, 1990.

> *SwitchFuel(DalhousieUnit#2 MintoCoal OffshoreCoal E3)*
> *time(E3 January13,1990)*

4. E4 is an event in which the fuel in Dalhousie Unit #2 was switched from Minto Coal to Offshore Coal. E4 occurred September 23, 1990.

> *SwitchFuel(DalhousieUnit#2 MintoCoal OffshoreCoal E4)*
> *time(E4 September23,1990)*

5. E5 is an event in which Dalhousie Unit #2 is brought online with Minto Coal as the fuel. E5 occurred July 15, 1990

> *PlantStartup(DalhousieUnit#2 MintoCoal E5)*
> *time(E5 July15,1990)*

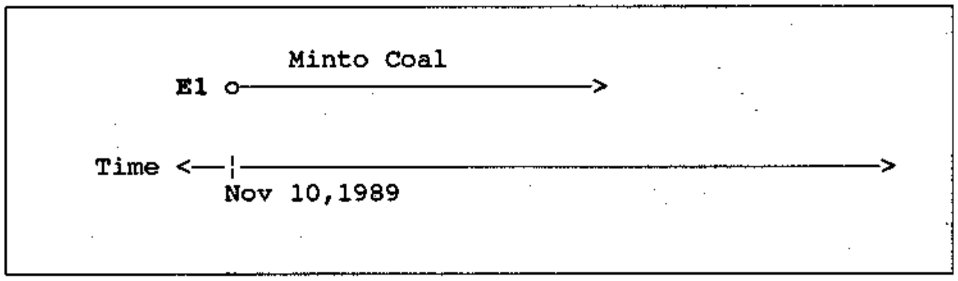


Figure 2-3 Update #1 Dalhousie Unit #2

After update #1, Figure 2-3, the fuel being burned at Dalhousie Unit #2 is Minto Coal. The time period after(E1) with its associated relationship, *FuelBurn(DalhousieUnit#2 MintoCoal after(E1))*, is initiated by event E1 and persists indefinitely in time.

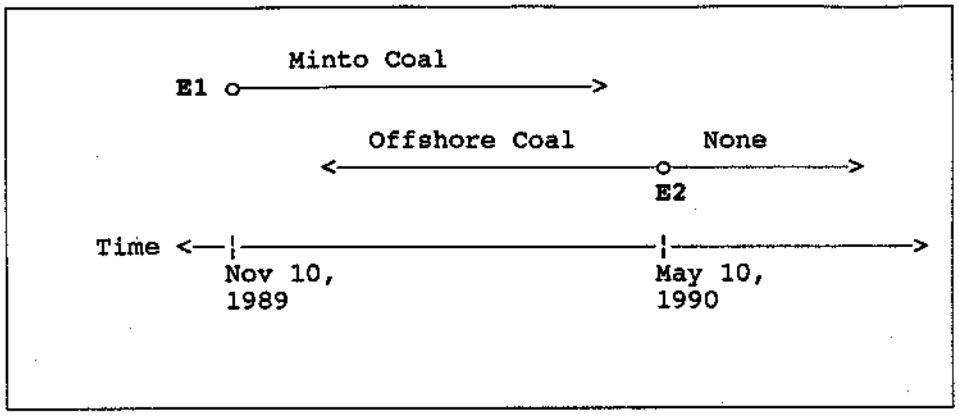


Figure 2-4 Update #2 Dalhousie Unit #2

E2 terminates the time period after(E1) initiated in Update #1. Analysis of Update #2 indicates that as of May 10, 1990 no fuel was being burned at Dalhousie Unit #2. A new time period before(E2) has been initiated where the associated relationship is *FuelBurn(DalhousieUnit#2 OffShoreCoal before(E2)).* From this it can be determined that Minto Coal was not the fuel burned up to the occurrence of E2, rather an event must have occurred between E1 and E2 which initiated the burning of Offshore Coal.

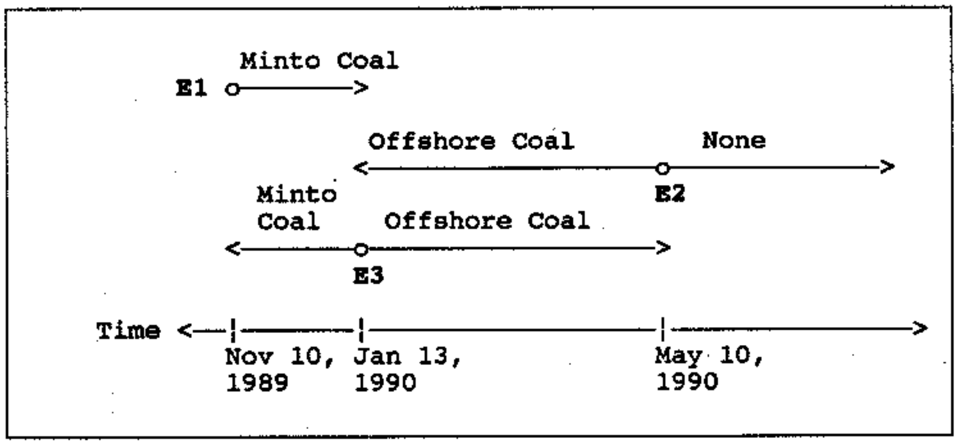


Figure 2-5 Update #3 Dalhousie Unit #2

Update #3 verifies the assumption made in the analysis of Update #2 that an event was not processed between E1 and E2. According to event E3, the fuel burned was switched on Jan 13, 1990 from Minto Coal to Offshore Coal. This initiated a new time period after(E3). Event E3 terminated the time period after(E1) and the time period before(E2). Event E2 terminates the time period after(E3).

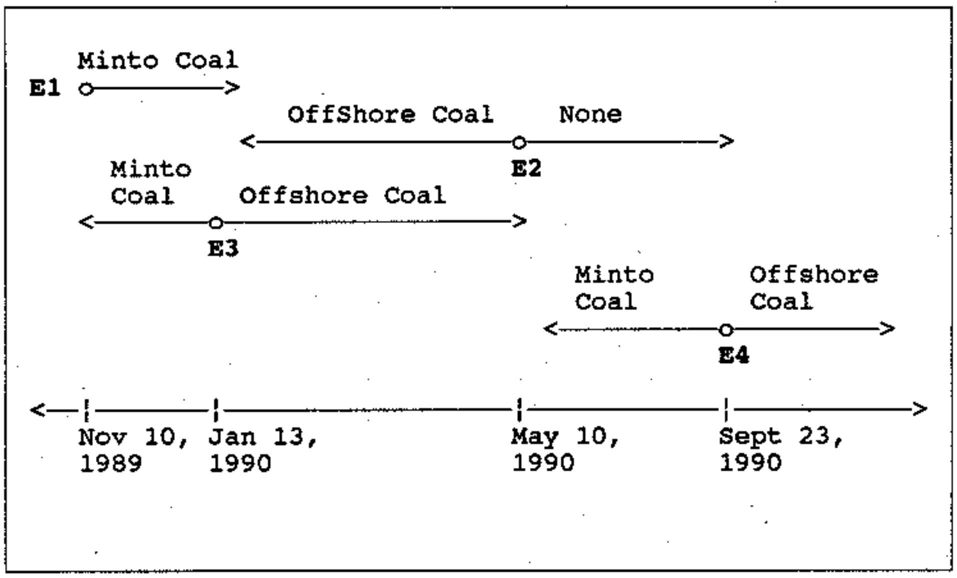


Figure 2-6 Update #4 Dalhousie Unit #2

Event E4 states that on Sept 23, 1990 there was a switch in fuel from Minto Coal to Offshore Coal. At this point in time no event has been processed which initiates the burning of Minto Coal. From this update it is known that the time period after(E2) is terminated some time before September 23, 1990. A new time period is initiated, after(E4) with its associated relationship

FuelBurn(DalhousieUnit#2 OffshoreCoal after (E4)).

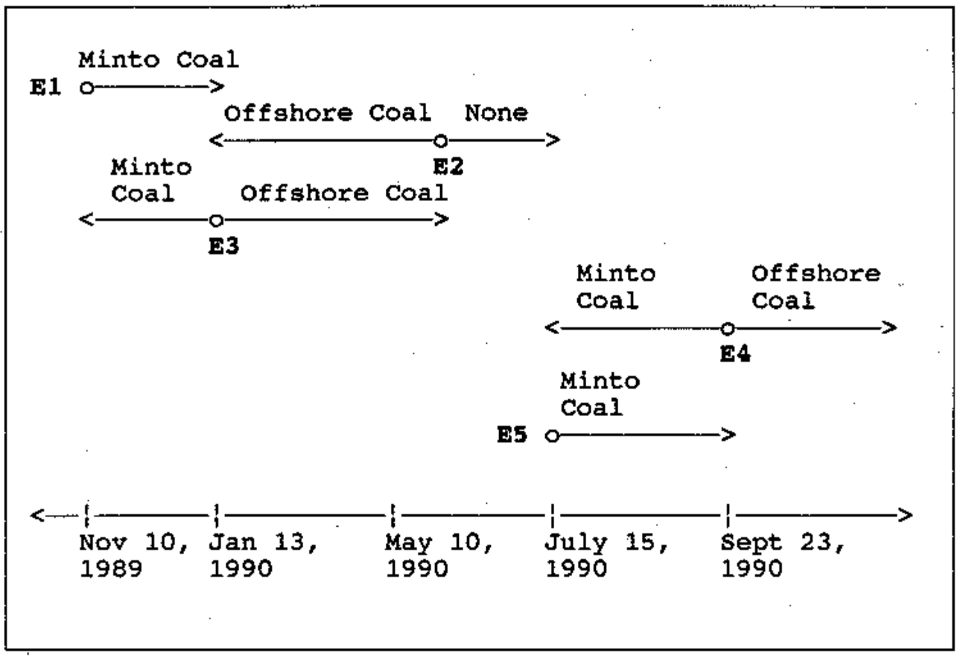


Figure 2-7 Update #5 Dalhousie Unit #2

Finally, event E5 is processed and is the event which initiates the time period after (E5) and the relationship *FuelBurn(DalhousieUnit#2 MintoCoal after (E5)).* As a result after(E2) is terminated along with its associated relationship. From previous updates it can be deduced that after(E5) is terminated by event E4.

Given all the events processed, the fuelburn relationship is summarized in Figure 2-8.

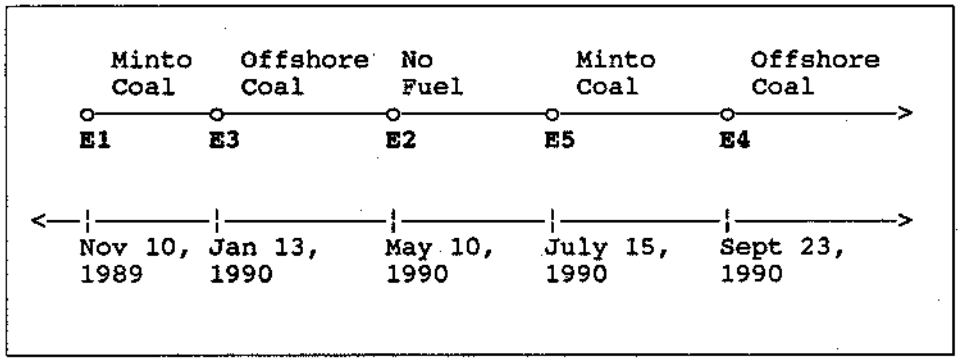


Figure 2-8 FuelBurn Relationship Summary Dalhousie Unit #2

This example illustrates some of the basic characteristics of event calculus. All updates are additive, as seen by the results of the successive updates. Information is never deleted. The only information that changes are the time periods initiated and terminated.

Past and future are treated symmetrically and thus events can be processed in any order. This is illustrated by update #3 where event E3 is processed after event E2 but which occurs in time, prior to E2. As discussed later in this chapter symmetrical treatment of time is handled with the use of the time periods after(e) and before(e). Events are ordered relative to each other. In event calculus it is not always necessary to associate explicit times with events, although in the FuelBurn example explicit times were used.

It is possible for actions and thus events to have duration. Time periods do not have to contain wholly the events which start or end them. For example, an event can be large enough that it can be divided into subevents with each subevent having associated time periods and relationships.

In order to justify the conclusions illustrated in Figure 2-8 it is necessary to prove that the following is true about the time periods in the example:

```
after( E1 ) = before( E3 )
after( E3 ) = before( E2 )
after( E2 ) = before( E5 )
after( E5 ) = before( E4 )
```

The following rules are used to prove the equality of time periods for the fuelburn relationship.

```
1.   The fuel burned during the time period t1 is
equal to the fuel burned during the time period t2.

2.   Chronologically, time period t1 starts before
time period t2 ends.
```

```
3.   No event occurs which changes the fuel being
burned after the start of the first time period t1
and before the end of the second time period t2.
```

In order to conclude:

```
end( after( E1 )  E3 )
and start( before( E3 )  E1 )
and end( after( E3 )  E2 )
and start( before( E2 )  E3 )
and end( after( E2 )  E5 )
and start( before( E5 )  E2 )
and end( after( E5 )  E4 )
and start( before( E4 )  E5 )
```

The following rules are presented by Kowalski:

```
end( after( e )  e' )
if after( e ) = before( e' )

end( before( e' )  e )
if after( e ) = before( e' )
```

In event calculus the concept of disjoint periods is required and is defined as follows:

```
p1 << p2
if end( p1 e )
and start( p2 e' )
and e < e'
p=[after( e ), before( e' )]
```

Also required is the axiom which states that two periods associated with the same relationship are either identical or they are disjoint. Relative to the example presented, this means that two fuels cannot be burned at the same time in the same plant. Given this axiom, Kowalski presents the following rule based on the *FuelBurn* example:

```
after( e ) = before( e' )
if FuelBurn( x y after( e ) )
and FuelBurn( x y before( e' ) )
and e < e'
and not after( e ) << before( e' )
```

As discussed by Kowalski, the definition of start and end is incomplete; thus it sometimes becomes difficult to prove that two time periods are disjoint. The incompleteness issue does not arise here due to the fact that events occur chronologically.

A redefinition of disjoint results in the following rule:

```
after( e ) = before( e' )
if FuelBurn( x y after( e ) )
and FuelBurn( x y before( e' ) )
and e < e'
and not [ FuelBurn( x y') after( e* ) and e < e*
and e* < e' ]
and not [ FuelBurn( x y') before( e* ) and e < e*
and e* < e' ]
```

Again, referring to the *FuelBurn* example, if it is necessary to determine the type of fuel being burned at a particular instant of time, Kowalski has given the following rule:

```
FuelBurnAt( x y t )
if FuelBurn( x y p )
and t in p.

t in p
if start( p e1 ) and end( p e2 )
and time( e1 t1 ) and time( e2 t2 )
and t1 < t and t < t2
p = [ after( e ), before( e )]
```

2.3 Simplifications Of Event Calculus

Kowalski presents a simplification of his event calculus based on a special case scenario found in some problem domains. This scenario is characterized as follows:

1. Events occur and are processed in chronological order.

2. The database must contain a complete record of all relevant past events. This is needed to guarantee all events have been processed to obtain the current state.

3. All information about an event is complete.

4. An event can only start and end one relationship.

When the problem domain of sootblowing advice in a coal-fired generating station was examined, the above characteristics were identified. Due to the nature in which process I/O is obtained, chronological ordering of the data is guaranteed. The temporal history mechanism implemented in the SAX system provides the complete record of all events. For this thesis, the event information was complete and a one to one relationship between an event and a temporal relationship was maintained. For these reasons the simplified event calculus proposed by Kowalski was adopted for this thesis.

Since all events which occurr are processed in order it is guaranteed that a relationship will be derived first as *FuelBurn(x y after(e))* before it is redundantly derived as *FuelBurn(x y before(e'))*. This removes the need for the processing of before(e') time periods. The resultant temporal predicates are *FuelBurn(x y after(e)),* *start(after(e) e)*, and *end(after(e) e')*

In the simplified case, it is also appropriate to say that a relationship holds at time t if it is started and t has not ended.

```
t in p
if start( after( e ) e )
and time( e t')
and t' <  t
and not end( after( e ) e' )
```

2.4 Event Calculus - The General Case

In the general case it is convenient to associate relationships directly with time periods denoted as *after(e u)*, where e is the event initiating the time period and u represents the relationship associated with the time period. A general predicate which expresses that the relationship associated with the time period p holds for the time period p is *holds(p)*. As an example

holds(after(e1 FuelBurn(DalhousieUnit#2 MintoCoal)))

represents the time period after (e1) in which the specified relationship *FuelBurn* holds.

To prove that the holds predicate is true the following rule is used:

```
holds( after( e u ) )
if initiates( e u )
```

The initiates predicate is proven based on rules specific to the relationship u. The ending of a relationship in the general case can be expressed in the following way:

```
end( after( e u ) e' )
if holds( after( e u ) )
and holds( after( e' u') )
and e < e'
and not ( after( e* u" ) e < e* < e' )
```

Since all events occur in order and all events are reported this rule can be simplified to

```
end( after( e u ) e' )
if holds( after( e u ) )
and initiates( after( e' u' ) )
```

Figure 2-9 illustrates the successive updates from the previous FuelBurn example as they would now be processed given the simplifications made to the event calculus. In this figure, events are processed in chronological order.

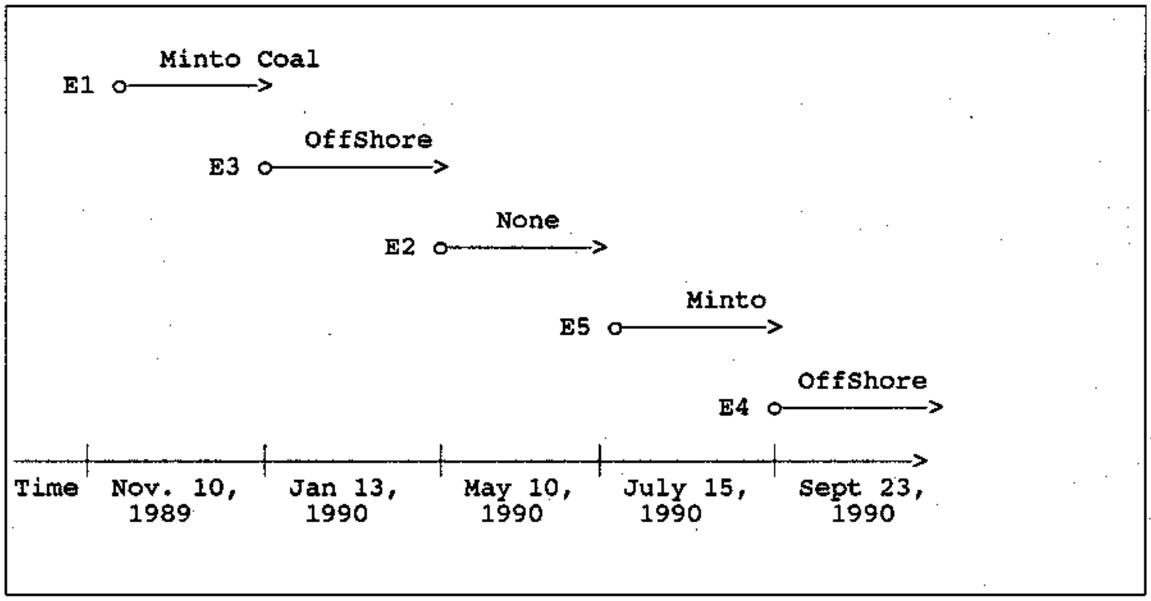


Figure 2-9 FuelBurn Updates Using the Simplified Event Calculus

3 IMPLEMENTATION OF THE TEMPORAL LOGIC - EVENT CALCULUS

In this chapter the implementation of the temporal logic event calculus will be discussed. Two approaches were explored. Each approach will be presented and a summary of the advantages and disadvantages given. The temporal logic was implemented as a preprocessor to SAX since all input data must first be processed temporally before it can be utilized by the SAX knowledge base.

3.1 Temporal Data

Before the temporal preprocessor can be implemented there first must be an understanding of the data about which it will have to reason. As seen in chapter 2, time periods are initiated and terminated. These time periods have an associated relationship. A relationship represents the state of a temporal object over the time period in which it was initiated. In the SAX problem domain there are two types of data. The first type of data is real-time process I/O which consists of analog values that are scanned every five seconds. This data must be captured by the temporal preprocessor and assimilated before the next I/O scan begins. If the temporal preprocessor is slow, data may be lost, resulting in a possible loss of vital information. As discussed in Chapter 1, SAX is classed as a soft real-time system and thus a response time of more than five seconds is not considered detrimental to the correct operation of the system. The second type of data is operator generated data. This data is entered by the operator through the SAX user interface.

On initial examination of the problem domain the following list of process I/O data was compiled:

1. Reheat Steam Temperature
2. Main Steam Temperature
3. Gas Exit Temperature
4. Main Steam Pressure
5. Excess 02
6. Steam Flow
7. Burner Tilt
8. Reheat Steam Attemporating Spray
9. Main Steam Attemporating Spray
10. Heat Absorption - Waterwalls
11. Heat Absorption - Reheater
12. Heat Absorption - Secondary
13. Heat Absorption - Primary
14. Heat Absorption - Economizer

The list of operator generated data was determined to be:

1. Boiler Observations
2. Mills In Operation
3. Current Fuel Type
4. Sootblowers Operated

The following two sections discuss each of the data types and how they are reasoned about in the temporal preprocessor.

3.2 Process I/O Data

For the purpose of the temporal preprocessor it was necessary to determine what relationships must be monitored for the process I/O variables. In general an analog variable value can *increase, decrease* or remain *steady* each time it is sampled, relative to the previous sample. A variable can be further classified based on the position of the value in the range of the variable; that is high, normal or low. High indicates that the value is high relative to the setpoint. Normal specifies the value is within an acceptable range of the setpoint. Low represents the value is below an acceptable distance from the setpoint. For each variable it is necessary to determine the limits of the normal range, thus delimiting the range of the high and low regions of the variable. Combining the variable characteristics the following relationships were derived:

1. Increasing High
2. Steady High
3. Decreasing High
4. Increasing Normal
5. Steady Normal
6. Decreasing Normal
7. Increasing Low
8. Steady Low
9. Decreasing Low

Figures 3-2, 3-3 and 3-4 pictorially represent the low, normal, and high temporal relationships respectively for the process I/O variables.

The variables were divided into three types as follows:

1. Setpoint variables
2. Process variables
3. Control device variables

A setpoint variable must be maintained at a given value called its setpoint which is accomplished using a control device. The correction made by the control device to the variable to maintain it at setpoint is determined by the error. The error is calculated as *(setpoint - actual value)*. Figure 3-1 pictorial represents the setpoint variable logic. A **setpoint variable** can be reasoned about based on the error. The magnitude, sign and slope of the error indicates the size of the error, whether it is above or below the setpoint and the direction of the change.

Since each variable specified has a possibility of representing different units and different value ranges, a common method of internal representation was required . This was accomplished by normalizing variable data as a percentage of its engineering range.

A normalized variable is calculated as follows:

$$\textit{normalized value} = \frac{\textit{variable value}}{\textit{(high range-low range)}}$$

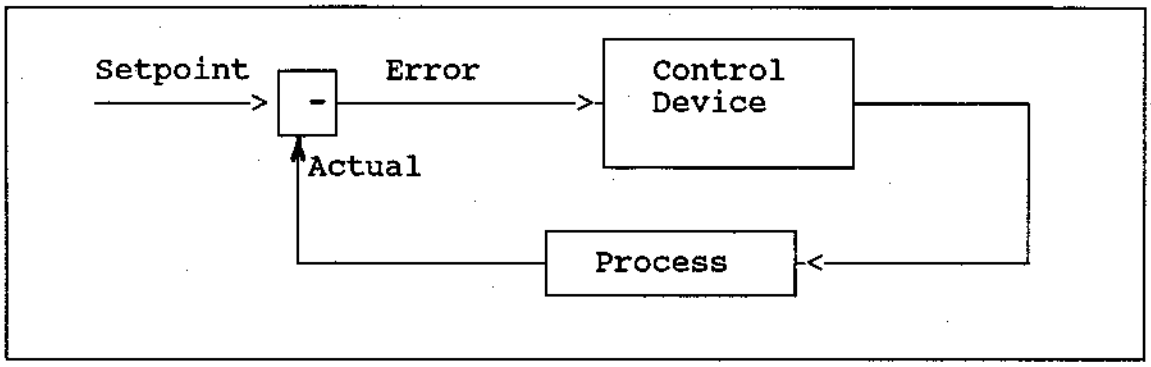


Figure 3-1 Setpoint Variable Control

Process variable values can vary over their full engineering range. There is no control mechanism available to maintain these variables at a setpoint value. From information acquired during the knowledge acquisition phase, these variables have values at which they should be maintained to achieve efficient boiler operation. These values are based on load conditions. For this thesis, a decision was made to limit the scope of the knowledge base by handling only the full load plant state. In so doing, it was possible to identify the acceptable values for these process variables. These values were established by the plant operating staff and are the optimum value at which the variable should be maintained to achieve efficient boiler operation during full load plant conditions. For this thesis these established operating values will be referred to as the internal setpoints. Since process variables now have defined setpoints they can be processed in the same fashion as setpoint variables.

For **control device variables** a true setpoint value cannot be established. It is still possible, however, to set a value about which temporal reasoning can occur. This value will be called the control device setpoint.

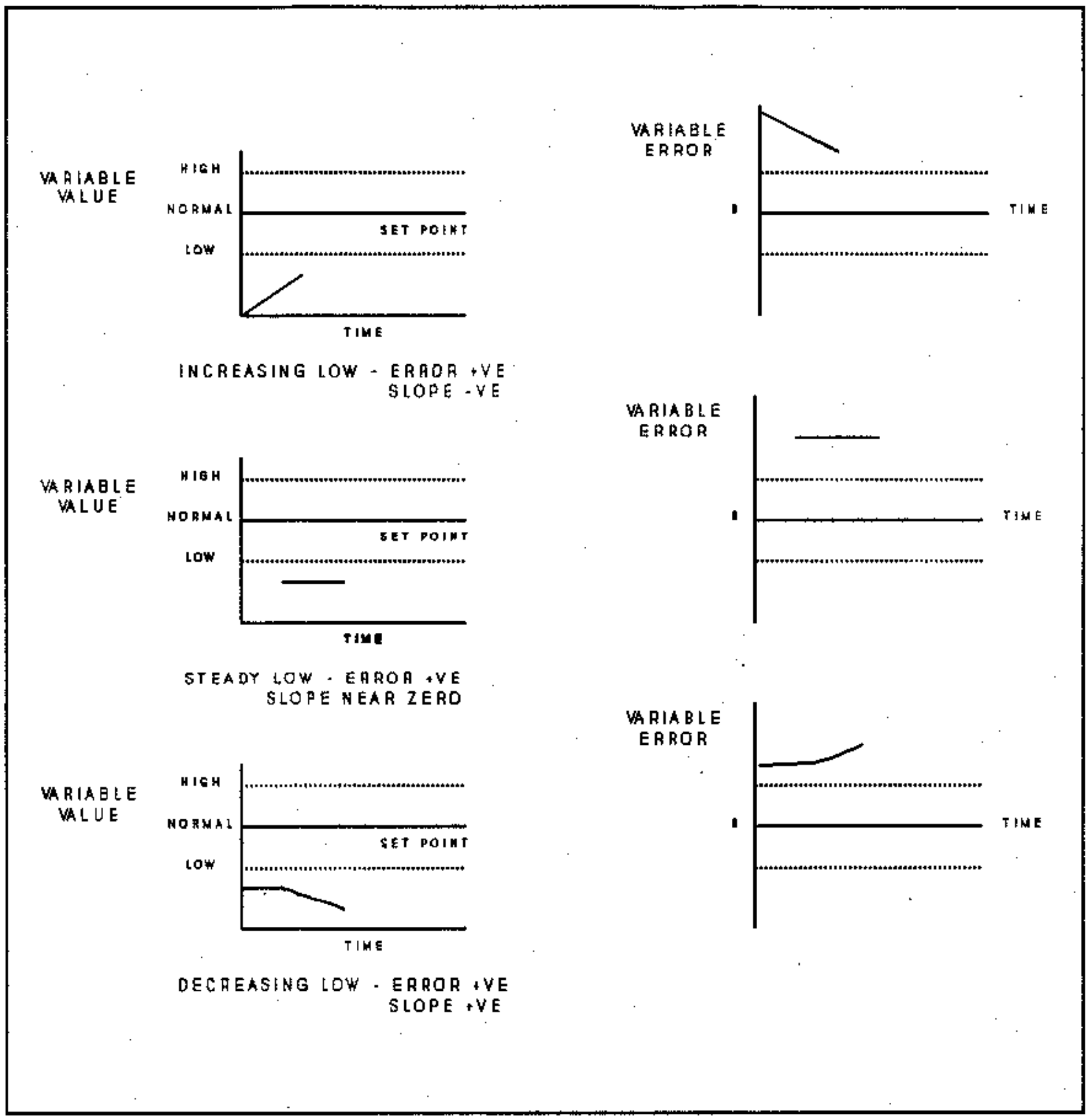


Figure 3-2 Low Variable Temporal Relationships

By fitting the control device variables into the setpoint variable model it is possible to use the same mechanism for temporal reasoning established previously for setpoint variables.

For Burner Tilt, which is a control mechanism for steam temperature control, the desired information is the position of the tilt relative to the horizontal position. There is no established normal range for burner tilt operation. Since burner tilt is reported

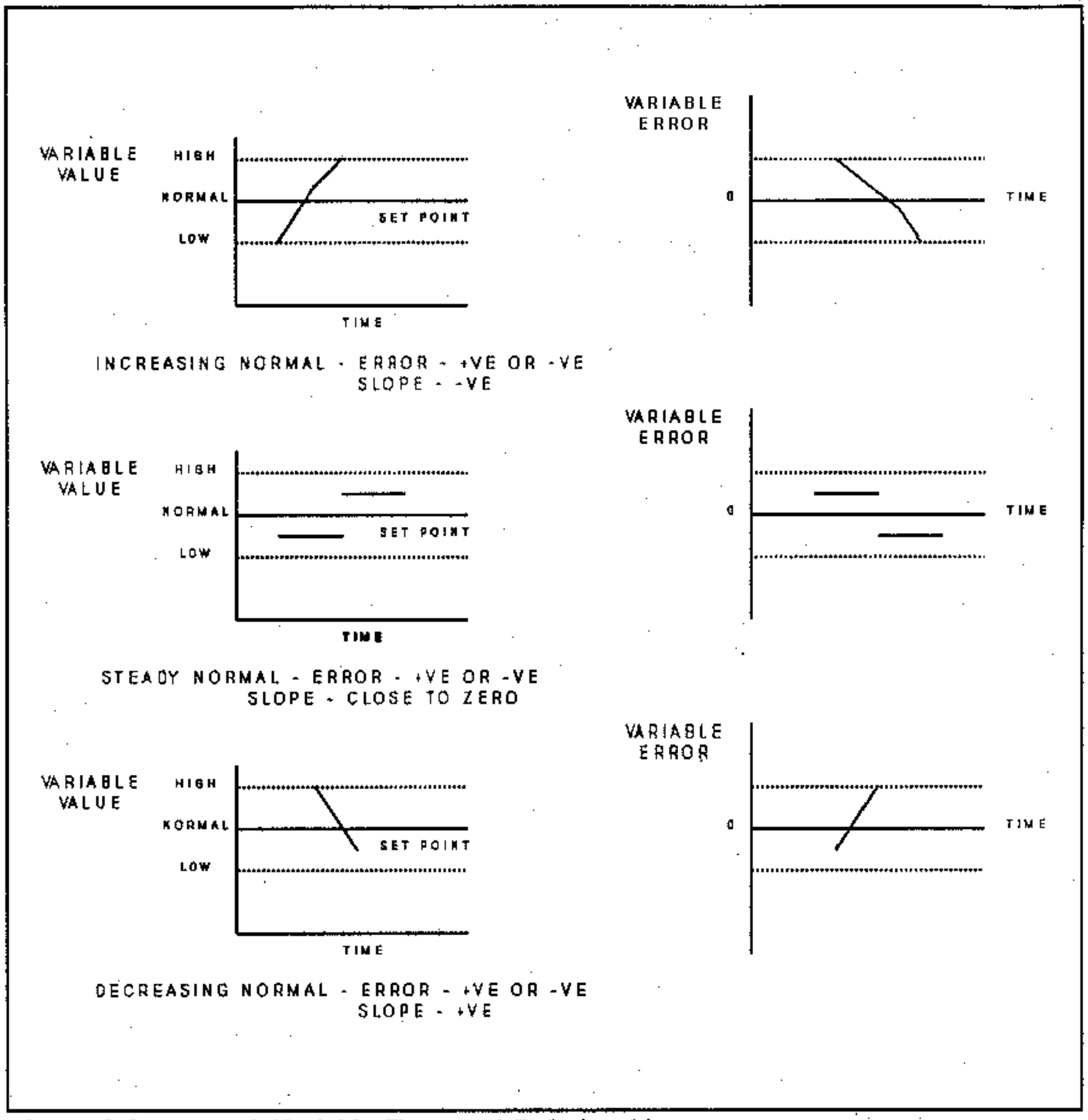


Figure 3-3 Normal Variable Temporal Relationships

as a percentage of its signal range, by setting the normal range value to 0 and the setpoint at 50, where 50 is the horizontal position the following variable relationships are valid:

1. increasing high
2. decreasing high
3. steady high
4. steady normal (horizontal)
5. increasing low
6. steady low
7. decreasing low

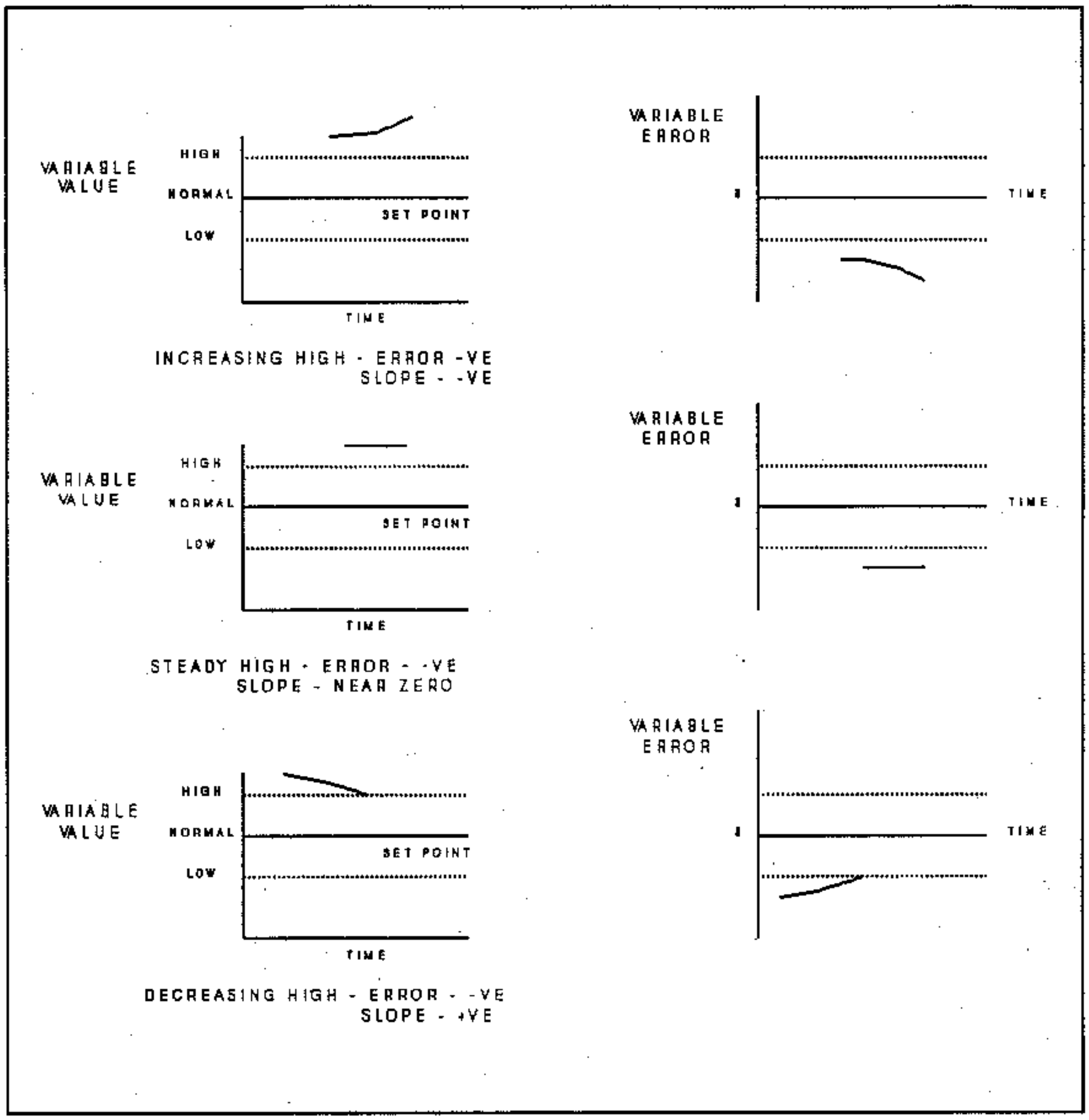


Figure 3-4 High Variable Temporal Relationships

Reheat and main steam attemporating sprays are also used to control steam temperatures. Optimally, they should not be used as they reduce the efficiency of the boiler. By setting the normal range of operation to 0 and the setpoint to 0, where the sprays are measured in lbs/hr, we establish the following temporal relationships:

```
1.  increase high
2.  decrease high
3.  steady high
4.  steady normal (no sprays)
```

Table 3-1 has the established variable data which will be used by the temporal processor to establish temporal relationships for the variables specified.

Variable	Units	Type	Setpoint at full load	Low Engineering Range	High Engineering Range	Normal Range
Main Steam Temperature	Fahrenheit	Setpoint	1000°	0	1600	±10
Reheat Steam Temperature	Fahrenheit	Setpoint	1000°	0	1600	±10
Gas Exit Temperature	Fahrenheit	Process	660°	0	1600	±10
Main Steam Pressure	PSIG	Setpoint	1800	0	2000	±10
Excess 0_2	%	Process	2.3	0	10	±0.2
Steam Flow	lb/hr (x 10,000)	Process	140	0	150	±0.5
Burner Tilt	%	Control	50	0	100	0
Main Steam Attemp. Spray	lb/hr (x 1000)	Control	0	0	30	0
Reheat Steam Attemp. Spray	lb/hr (x 1000)	Control	0	0	30	0

Table 3-1 Process I/O Variable Data

3.3 **Operator Generated Data**

There are static temporal objects which must be reasoned about in the temporal preprocessor. For these objects temporal relationships must be developed. Along with this the rules to initiate and terminate the relationship must be established. The temporal objects types are categorized in the following way:

1. Sootblowers
2. Boiler Observations
3. Fuels
4. Mills

Each of the above temporal object types contain multiple temporal objects. For instance in the sootblowers there are approximately ninety individual sootblowers. The boiler observations are broken down into boiler sections such as the waterwall. There are approximately one hundred boiler sections identified. There are four possible fuels types identified and four mills that can be operated at the plant.

3.3.1 **Sootblower Temporal Relationships**

Two temporal relationships have been identified for sootblowers. They are ***Available** for operation by the operator* and ***Unavailable** for operation by the operator.* These relationships are based on the criteria that a recently operated sootblower should not be operated again for a period of eight hours. Frequent operation of the same sootblowers causes corrosion damage to occur around pipes, fittings, etc. Also, if a sootblower has been operated in the last eight hours it would be more effective to operate alternate sootblowers when removing detected slag.

3.3.2 **Boiler Temporal Observations**

Boiler observations are used to assess the slagging conditions in the boiler. The buildup of slag in the boiler is classified according to the type and amount of buildup. The type of buildup can be either dust or slag. The amount of buildup is measured in inches and can be classified as follows:

```
1.    0"              - no buildup
2.    0"-1.5"         - low buildup
3.    1.5"-3.5"       - medium buildup
4.    3.5" and up     - high buildup
```

Combining the characteristics of type and amount the following temporal relationships were identified:

```
1.     Dust none
2.     Dust low
3.     Dust medium
4.     Dust high
5.     Slag none
6.     Slag low
7.     Slag medium
8.     Slag high
```

3.3.3 Fuels and Mills Temporal Relationships

Fuels are simple temporal objects which are either **Selected** or **Deselected.** A fuel which is selected is the fuel currently being utilized at the plant. Only one fuel can be selected at a time.

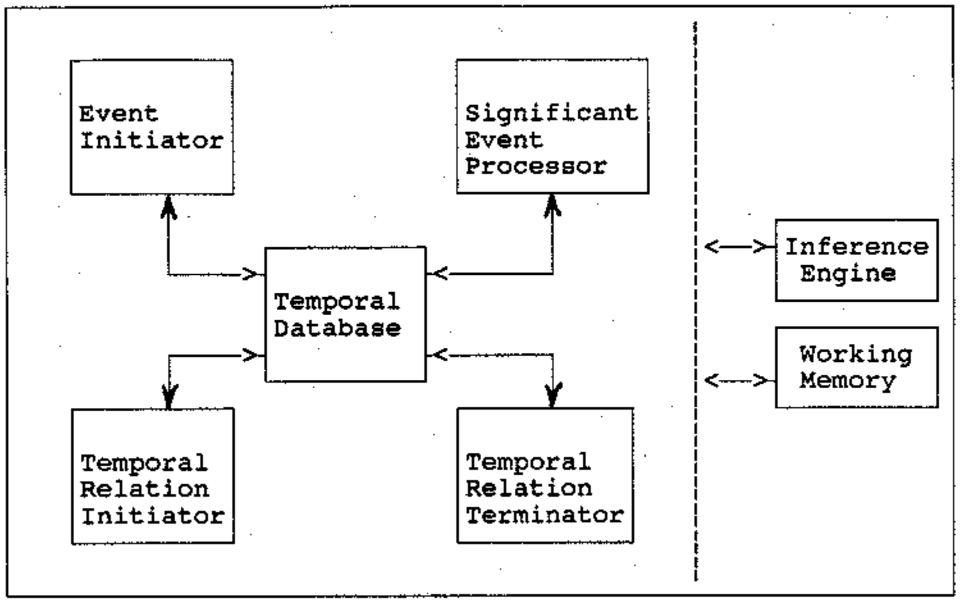
Mills are also simple temporal objects which are either **Selected** or **Deselected.** A mill which is selected is currently operational whereas a mill which is deselected is not operating in the plant.

3.4 Overview of the Temporal Preprocessor

The temporal preprocessor can be logically divided into the following components:

```
1.     Event Initiator
2.     Significant Event Processor
3.     Temporal Relation Initiator
4.     Temporal Relation Terminator
```

Figure 3-5 pictorially represents how these components are organized to form a temporal preprocessor. Execution of the temporal preprocessor results in a temporal database upon which the SAX knowledge base can make decisions about sootblowing strategies.


Event Initiator
Significant Event Processor
Temporal Database
Inference Engine
Working Memory
Temporal Relation Initiator
Temporal Relation Terminator
Figure 3-5 Temporal Processor

The event initiator obtains raw event data about a temporal object and creates an event descriptor. The event descriptor contains information about the event such as the temporal object, the time of occurrence of the event and other event specific data.

The significant event processor is a set of rules which determines if an event is significant. An event is significant if it results in a change in the current relationship of a temporal object. An event which is not significant is deleted from the temporal database as it serves no useful purpose.

The temporal relation initiator starts a temporal relationship for a temporal object based on a significant event. The temporal relation terminator ends a temporal relationship and is invoked when there are two temporal relationships in existence for one temporal object. When a new relationship is generated for a temporal object, the old relationship automatically becomes obsolete and must be terminated. Termination of a temporal relationship automatically activates the temporal history processor which will be discussed in the next chapter.

3.5 Prolog Temporal Preprocessor

The SAX system was an idea conceived as a result of the Improved Steam Temperature Control Real-Time Expert System being developed at Maritime Nuclear. In this system the Turbo C procedural language was used in conjunction with the data driven Turbo Prolog language to provide a combination approach to the expert system solution. Since these software tools were already in place, Turbo Prolog was chosen as the first software tool in which to implement the temporal preprocessor. By choosing this language the steam temperature control simulator already developed as part of the Steam Temperature Control Real-Time Expert System was made available for testing of the temporal preprocessor.

To test the feasibility of using Turbo Prolog to implement the temporal preprocessor a subset of the problem was chosen. This included the processing of two variable temporal objects, reheat steam temperature and main steam temperature, which was a small subset of the actual number of temporal objects required.

3.5.1 Data Acquisition

Variable data is generated periodically by the Steam Temperature Control Simulator. The availability of this data is signalled to the temporal preprocessor via a call to a Turbo Prolog predicate *pass_temporal_vars*. This predicate then calls Turbo C based routines to obtain the variable data. For example *c_get_reheat_vars* was called from *pass_temporal_vars* to obtain the reheat steam temperature data.

For the above operation to successfully take place a complicated set of procedures for Turbo Prolog to Turbo C language interfacing were followed [PROL88]. An example of one of these rules involves the complications which occur when the two languages are linked to form an executable program. Normally when a Turbo C program is compiled all unresolved references are preceded by an underscore "_". During the linking phase of the Turbo C compiler these references are resolved. When linking a Turbo Prolog program, however, these underscores have no meaning and the references remain unresolved. For this reason a Turbo C compiler option is used to inhibit the generation of underscores during compilation. A second stumbling block occurred when parameters were passed between the two languages. To minimize these problems simple data types were used when passing parameters between Turbo Prolog and Turbo C.

3.5.2 **Data Structures**

The implementation of the temporal preprocessor began by designing the data representation for event calculus. A Turbo Prolog database was defined to maintain the temporal database. The temporal predicates were defined as follows:

```
holds( relation_desc )
start( time_period )
end( time_period )
event_in( event_list )
```

The *holds* predicate is asserted into the database when a relationship is initiated. The relationship is identified by the *relation_desc*. The *start* predicate is asserted to associate a time period with the relationship. The *end* predicate is asserted when a temporal relationship is terminated. The *event_in* predicate maintains the list of valid event descriptors. To support these predicates the following compound data structures were defined:

```
time_period      = after( relation_desc )

relation_desc    = relation_desc( relation_type,
                                  object,
                                  event_key )
```

```
event_list          = event_desc*

event_desc          = event_desc( eventkey,
                                  object,
                                  act,
                                  event_date_list,
                                  init_relation,
                                  term_relation )
```

3.5.3 **Event Initiator**

Once the temporal data is acquired by the temporal preprocessor it is stored in a temporary raw data format. The *create_event* predicate is called and matches the data from the raw temporal database, retracts the data and via the *create_event_desc* predicate generates an event descriptor. An event key is generated for the event descriptor which together with the event time uniquely identifies the event. The event is then added to the event list, *event_in* found in the temporal database. This list maintains all the events currently active in the temporal database.

3.5.4 **Significant Event Processor and Temporal Relation Initiator**

The significant event processor is initiated by a call to *process_temporal_events* which processes new events from the event_in list found in the temporal database. For each temporal object type the possible relationships that can be started have been previously identified. A relationship predicate is asserted for each possible relationship of the temporal object. For example, the temporal relationships for the reheat steam temperature temporal object which can be initiated are increase_high, increase_steady, decrease_low etc (see chapter 2). The significant event processor matches an initiation rule against each relationship predicate which has been asserted.

If the temporal initiation rule for the temporal object and its associated relationship is successful then a *holds(relation_desc)* is asserted into the temporal database. A start predicate *start(after(relation_desc))* is also asserted into the temporal database.

3.5.5 **Temporal Relation Terminator**

The temporal relation terminator asserts an *end(after(relation_desc))* into the temporal database when a relationship is terminated. This is invoked when a new relationship is initiated since the initiation of a new relationship automatically means the termination of the old relationship for that temporal object. A comparison of the event times for the temporal relationships determines which relationship is terminated.

3.5.6 **Results**

From this implementation of the temporal preprocessor in Turbo Prolog, simple tests were performed to initiate and terminate relationships. These tests and results were limited as a result of complications which occurred during testing (see the next section for details).

3.5.7 **Advantages and Disadvantages of the Prolog Implementation**

The advantages of the Turbo Prolog implementation of the temporal preprocessor are as follows:

1. Kowalski presents his event calculus in a predicate calculus form and gives implementation details for the Turbo Prolog language.

2. There is a simulator in Turbo C available to properly test the temporal preprocessor.

The disadvantages of the Turbo Prolog implementation are as follows:

1. The Turbo C - Turbo Prolog language interface is cumbersome and very prone to error.

2. Because of the inter-language linkage there are no debugging facilities, thus making testing and verification of the system impossible.

3. Turbo Prolog is a backward chaining inference engine. The data driven nature of SAX lends itself more to a forward chaining inference engine.

4. A physical memory limitation was reached when linking the two languages even with the small subset of the temporal preprocessor.

3.6 ART-IM Temporal Preprocessor

A second implementation of the temporal preprocessor was developed using the DOS based expert system tool, the **Automated Reasoning Tool** by Inference Corporation. Some of the features of ART-IM are:

1. A forward chaining inference engine
2. An Interactive development environment
3. Provides extension and integration with 'C'
4. A user interface toolkit
5. It breaks the 640K barrier by accessing extended memory
6. Allows deployment of ART-IM applications as an executable program
7. Allows external data integration

Since ART-IM provided many features which were not available in **Turbo Prolog** a complete implementation of the SAX temporal preprocessor was realized. In this section, presentation of the developed temporal preprocessor will begin with Data Acquisition, followed by Data Structures, Event Initiator, Significant Event Processor, Temporal Relation Initiator and Terminator.

3.6.1 **Data Acquisition**

Data acquisition, as discussed previously, can be divided into two main components, process I/O data and operator generated data.

Process I/O

Process I/O data acquisition in the SAX system was handled differently than in the Turbo Prolog implementation of the temporal preprocessor. In Turbo Prolog a front end simulator written in Turbo C provided data periodically to the temporal preprocessor. For the prototype SAX system in ART-IM, process I/O data is acquired from an external data file and read periodically. This process I/O data is asserted as a fact in the factbase for processing by the temporal preprocessor. Formulation of the external data file will be discussed in detail in Chapter 6.

Operator Data

Operator generated data is acquired via the SAX user interface which was developed using ART-IM'S user interface toolkit. This data was then asserted as a fact in the factbase for processing by the temporal preprocessor. The implementation details of the SAX user interface will be discussed in Chapter 5.

3.6.2 **SAX Data Structures**

Within the SAX temporal preprocessor the different temporal objects were defined as schemas. A schema is a collection of information about a particular object stored in the database. It allows an object to be reasoned about as a whole or based on individual characteristics of the object. Two features of schemas which SAX takes extensive advantage of are (1) the ability to structure objects into related groups and

(2) the ability to organize these schemas into hierarchies in which knowledge about an object can be automatically deduced based on the class to which it belongs.

The following is a list of the groups of schemas which have been defined for the temporal objects.

```
1.    Mills
2.    Fuel
3.    Sootblowers
4.    Boiler Observations
5.    Variables
```

An example of how these are defined in a hierarchial scheme is illustrated in Figure 3-6. In this figure, all sootblowers are defined as plant sootblowers with common attributes. This is then divided into groups from superheater sootblowers to slope sootblowers. Each of these groups of sootblowers are subdivided accordingly. The waterwall sootblowers, as illustrated, are broken into three subgroups; lower, middle and upper. At the lowest level the individual sootblowers are defined, as shown, for the middle waterwall. In this hierarchial representation it is possible to reason about all sootblowers by using plant sootblower or the sootblowers defined under middle waterwall or at the lowest level, sootblower IR_13.

Inheritance of attributes automatically occurs as a result of the hierarchical schema structure. The plant sootblower schema is assigned a group of attributes. These attributes will be inherited by all objects in the hierarchy. In other words if the attribute *(Sootblower, State Operate)* is defined in the Plant Sootblower Schema, the schema IR_13 at the lowest level of the hierarchy would also have this attribute. Attributes defined in the waterwall schema will be inherited by the lower, middle and upper waterwall schema but not by the superheater schema. Figure 3-7 gives examples of schema definitions for each level of the sootblower hierarchy after inheritance has automatically propagated the attributes.

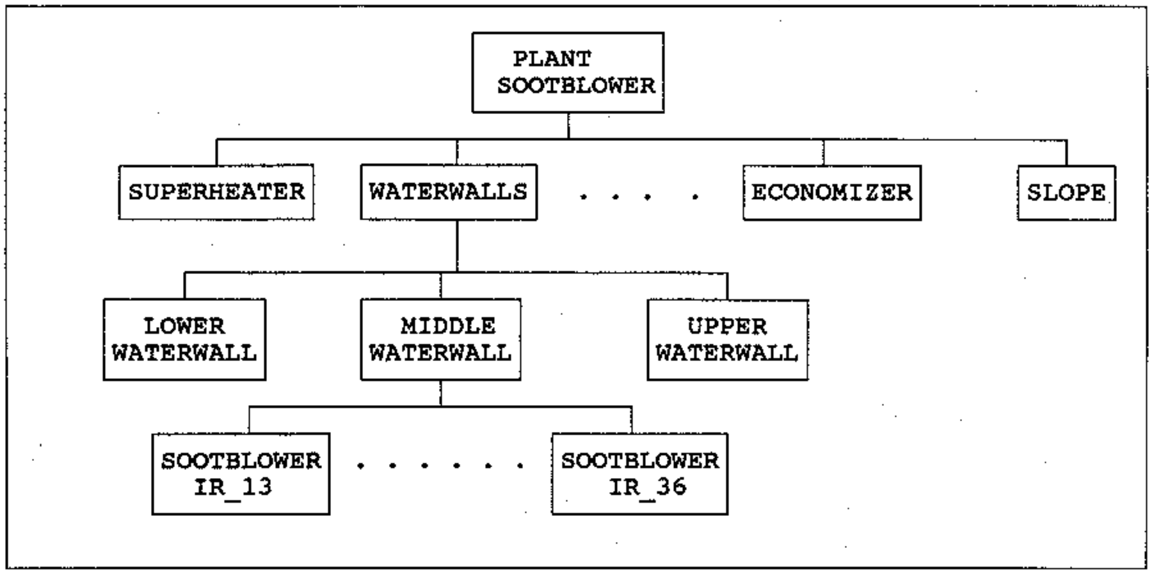


Figure 3-6 Schema Hierarchy for Sootblowers

A frame based knowledge representation was chosen because the organization of data to be incorporated into the system was object oriented in nature. Secondly, it was necessary to be able to reason about all, part or individual components of the hierarchy. The ability to use default knowledge allowed for automatic updates and consistency constraints in the knowledge base.

3.6.3 **Event Initiator**

The event initiator is divided into sections of rules where each section generates an event for a particular type of temporal object. As discussed earlier, the type of temporal objects included variables, mills, sootblowers, fuels and boiler observations.

```
( defschema plant_sootblower
        ( operation_time 5 )
        ( activate 28800 )
        ( operation_status inactive )
        ( soot_action nil ) )

( defschema waterwall_sootblower
        ( is-a plant_sootblower )
        ( type IR )
        ( area waterwalls )
        ( operation_time 5 )
        ( activate 28800 )
        ( operation_status inactive )
        ( soot_action nil ) )

( defschema middle_waterwall_sootblower
        ( is a waterwall sootblower )
        ( section above burners )
        ( type IR )
        ( area waterwalls )
        ( operation_time 5 )
        ( activate 28800 )
        ( operation_status inactive )
        ( soot_action nil ) )

( defschema IR_13
        ( instance-of middle_waterwall_sootblower )
        ( orientation mww_s_bcor )
        ( location south_b_corner )
        ( section above burners )
        ( type IR )
        ( area waterwalls )
        ( operation_time 5 )
        ( activate 28800 )
        ( operation_status inactive )
        ( soot_action nil ) )
```
Figure 3-7 Plant Sootblower Schema After Inheritance

For the purpose of event initiation the temporal object types were divided into two kinds, those that required manipulation of the event data and those in which the event data could be used directly to initiate an event. Temporal process variables fell into the first category which required data manipulation. There were two ways in which the data manipulation could be accomplished, as part of the event initiator or as a data processor external to SAX. Incorporating the data manipulation into the event initiator required taking the variable value, setpoint and time and performing the following:

```
1.    Normalization of the variable value and setpoint
2.    Calculation of the error
3.    Calculation of the slope of the error
4.    Calculation of the average slope
5.    Conversion of the time string to a unique number
      relative to 0:00 hours on Jan. 1, 1990.
```

The data calculations in each of the above items has been discussed previously in this chapter except for the time string. The time string is used for the chronological ordering of events and is received in one of two possible formats; an externally defined time stamp *YY-MM-DD/HH:MM:SS* or an internal ART-IM time stamp *MON JAN 05 09:31:59 1991.* The time stamp is converted to a numeric number which is calculated as the number of seconds from 0:00 hours on January 1, 1990.

Given the amount of numeric calculations required before the variable data was ready for processing by the event initiator, it was decided to make these calculations external to the ART-IM temporal preprocessor in a data preprocessor which will be discussed in Chapter 6. This decision is a result of timing tests performed using each of the methods. Including the numeric calculations in the event initiator caused a significant increase in the time required to process events.

To generate an event the event initiator attempts to match a fact in the factbase of the following format:

(DataType TemporalObjectType Object Data1....Data2....DataN TimeString)

If a match occurs, event specific data processing is done resulting in an event schema being asserted into the knowledge base having the following structure:

```
( schema     ?event
        ( instance of            event )
        ( event_type             ?etype )
        ( object                 ?object )
        ( time_str               ?time_str )
        ( time                   ?time )
        ( SPECIFIC ATTRIBUTES ?attr )
        ( processed              No ) )
```

The variable ?event is a unique event name. This uniqueness is guaranteed by using an ART-IM function called *gentemp*. All events are grouped in a simple schema hierarchy. The event_type attribute contains the temporal object type. The object attribute specifies the temporal object in the designated type. An example of an event_type is *variables* with *reheat steam temperature* being the specific temporal object. The time attributes, time_str and time, represent the time the event occurred in string form and numeric form respectively. Depending on the type of temporal object other attributes are specified which contain the event specific data obtained from the matched fact. The processed attribute is a flag which indicates whether the event has been processed by the significant event processor.

Once the event schema has been asserted into the knowledge base the function of the event initiator is complete.

3.6.4 **Significant Event Processor**

The significant event processor attempts to match on an event which has not been processed. As discussed above, processed is an attribute of an event schema. As in the event initiator, the significant event processor is divided into rule sections based on the temporal object type. Each section contains the specific rules for a temporal object which determine if the event being processed is significant. An event is significant if it changes the current relationship and thus the time period for the temporal object in question. It is possible that an event is processed which does not alter the current relationships or time periods. This event is insignificant and is automatically deleted from the event hierarchy and thus the temporal preprocessor.

There were two implementation methods investigated for the significant event processor. They were the query method and the relation method.

The query method begins by determining the relationship which will be generated by the event being processed irrespective of the current relationship of the temporal object. This relationship is called the new relationship. A query is then initiated to determine the current temporal relationship for the temporal object in question. On completion of the query the current relationship for the object is known. Given the new relationship and the current relationship several possible scenarios can occur. If the new relationship equals the current relationship the event is insignificant and the event will be deleted. If the new and current relationships differ the temporal relation initiator is invoked to create the new relationship.

In the relation method, every temporal relationship which may be generated has three possible scenarios where each of these is represented by a rule. These rules are summarized as follows:

1. If this is the first event processed for an object then determine the current relationship and invoke the temporal relation initiator to start the new temporal relationship.

```
2.  Determine the new temporal relationship.  Compare
this to the current relationship. If they are not
equal invoke the temporal relation initiator to create
the temporal relationship specified by the new
relationship.

3.  Determine the new temporal relationship.  If
the current relationship is equal to the new
relationship then delete the processed event as
it is insignificant.
```

Comparison of the two methods was based on the number of rules required, the levels of processing and the amount of schema processing. The query method required three levels of processing; obtaining the current relationship, generating the query, and processing the query. The relation method required one level of processing. In comparing the number of rules the query method required 19 whereas the relation method required 55. By increasing the number of rules the relation method decreased the amount of schema processing required. It was generally determined that the processing time for the relation method was marginally faster than the query method. The relation method was chosen because it provided a more modular stand alone implementation of the significant event processor. As a result the testing and maintenance of this component was simplified.

3.6.5 **Temporal Relation Initiator and Terminator**

Before discussion of the temporal relation initiator and terminator it is necessary to understand how the event calculus predicates holds, start and end are represented in ART-IM. Initially, each predicate was represented separately as a fact in the factbase denoted as follows:

(holds after (event relation))
(start after (event relation))
(end after (event relation))

However, it was determined that more efficient processing could be achieved if the components of the temporal logic were represented as attributes in a schema. The temporal time period, event, and relationship components as well as the start, end and holds predicates are represented as follows:

```
(Schema   ?temporal)
          ( instance-of        temporal_relation)
          ( start-event        ?s_event )
          ( start-hfile        ?s_hfile )
          ( object             ?object  )
          ( relation           ?relation)
          ( end-event          ?e_event )
          ( end-hfile          ?e_hfile ) )
```

In this schema *?temporal* is the unique name for the temporal relation and is created using ART-IM's *gentemp* function. The *temporal_relation* attribute specifies the class of objects for temporal relationships. The *start_event* and *end_event* attributes specify the event schemas which initiated and terminated the relationship respectively. The *start_hfile* and *end_hfile* attributes specify the history file schemas in which the start and end of the temporal relationship were recorded respectively. The specific temporal object for this temporal relationship is specified in *object*. The *relation* attribute contains the name of the relationship initiated for the object.

The temporal relation initiator is invoked as a function call from the significant event processor. It is passed the new relationship, temporal object and the initiating event. Given these, it asserts a new temporal relationship in a temporal relation schema as discussed above. The attributes of *end_event* and *end_hfile* are not asserted until the temporal relation is terminated.

The temporal relation terminator matches on pairs of temporal relationships for the same temporal object. If this situation occurs, it is known, based on past assumptions, that one of these relationships has terminated. The temporal relation to be terminated will be the one which is chronologically earlier in time. This is determined by using the time attribute in the initiating event schema. When a temporal relationship is terminated the *end_event* and *end_hfile* attributes are asserted into the temporal relation schema. The *end_event* recorded will be the event which initiated the new relationship.

3.6.6 Advantages and Disadvantages of ART-IM

Several advantages were realized when using ART-IM as the software tool. These advantages included:

1. Forward chaining inference engine
2. A knowledge base which is object oriented, frame-based with inheritance capabilities
3. Asynchronous processing
4. User Interface Toolkit
5. External Programming Interface

The disadvantages of using ART-IM occurs in the area of testing the SAX system. In the Turbo Prolog implementation there was a steam temperature control simulator which provided a test bed for the temporal preprocessor. By using ART-IM, an alternate method of testing was required. Given the insurmountable odds of a Turbo Prolog implementation as discussed previously in this chapter, it was determined that ART-IM would be the software development environment used for the complete implementation of SAX.

4 TEMPORAL HISTORY PROCESSOR

4.1 Introduction

As discussed in Chapter 2, one of the requirements of the simplified event calculus is that the temporal database contain a complete record of all relevant past events. This is needed to guarantee that all events have been taken into account to obtain the current state of the temporal database. To achieve this, a temporal history processor was implemented in SAX. To access the temporal history as well as the current temporal database, a query processor was also implemented. The temporal history processor is made up of the following components:

```
1.    Store History
2.    Restore History
3.    Query Temporal Databases (current and history)
```

Before these components are discussed it is necessary to present the file structure used for the accumulation of temporal history data, as well as the type and structure of the data which will be stored.

4.2 History Files

Temporal history data accumulation is significant because SAX is a real-time system where temporal relationships can change quickly. History files have the possibility of growing large rapidly. For this reason, temporal history is stored in files which have a maximum of three hour duration. An advantage of keeping history files relatively small occurs during the query cycle. A small file limits the amount of information which must be loaded into SAX for temporal history processing.

Each history file is identified in SAX by a class of objects called *history_file*, the basic format of which is identified in the following schema:

```
(schema   ?h_name
          (instance-of          history_file)
          (file                 ?h_file     )
          (stime_str            ?stime_str  )
          (stime                ?s_time     )
          (current              yes        ))
          (etime_str            ?e_str      )
          (etime                ?etime      )
```

In the history_file schema *?h_name* is the unique name identifying the history file schema. The *stime_str* contains the start time of the history file represented as a string. The *stime* contains the start time of the history file in a numeric form. *Current* is an attribute which indicates whether this is the current history file for temporal history accumulation. When a history file is closed for accumulation of temporal history the attributes *etime_str* and *etime* are asserted into the history file schema. These two attributes contain the time in which the history file was closed in string and numeric form respectively.

The opening and closing of history files is automated by using a timing mechanism. When a history file is opened an associated time fact is asserted into the factbase of the form *(history_time   history_file   end_file_time)*. The *history_file* specifies the history file schema and *endfile_time* contains the calculated time at which the history file should be closed and the next history file opened. The timer rule is triggered periodically but at a lower salience than other rules, where salience specifies the priority of rule execution. If there is no rule remaining on the SAX rule agenda with a higher salience then the timer rule will fire. In this rule, the *endfile_time* is compared to the current time. If it is greater, the file is closed and the current attribute is updated to No. A fact is then asserted to indicate the requirement of a new history file.

It is necessary for every history file to contain all the temporal information necessary to assess the temporal relationships contained within. The reason for this is illustrated in Figure 4-1.

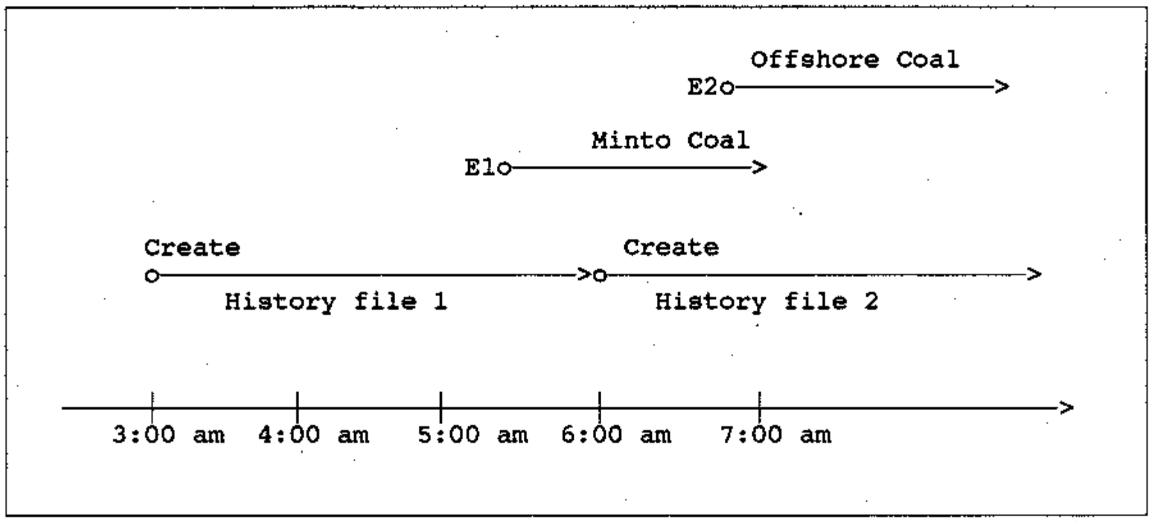


Figure 4.1 Temporal History and Temporal Relationships

In this figure we have history file 1 created at 3:00 a.m. Recorded in this file is the start of the time period *holds(after(e1 FuelBurn(DalhousieUnit#2 MintoCoal)))*. History file 1 is terminated at 6:00am and a new history file, history file 2 is created. In order for history file 2 to contain all the temporal information it must not only record the time period *holds(after(E2 FuelBurn(DalhousieUnit#2 OffshoreCoal)))* but also rerecord the time period relative to E1. As illustrated by this example, when a new history file is created the first thing that must occur is that all temporal relationships which are currently in existence be recorded in the history file. Any new temporal relationships should then be recorded as they are initiated. Temporal relationships which are terminated must also be recorded.

4.3 History File Records

As discussed previously when a new history file is created, the first step taken is to record all the current temporal relationships. Newly generated relationships are then recorded as they occur, as well as all terminated relationships. There are three basic types of records which are used in history files. They are the history start record, history end record, and history event record. In each record the first field identifies the record type. A history start record has the following format:

```
( def_external_data  history_start_record
        ( rec_type              0  3: char  : symbol )
       *( s_name                3 20: char  : symbol )
        ( object               23 15: char  : symbol )
        ( relation             38 15: char  : symbol )
        ( start_event          53 10: char  : symbol ))
where
        rec_type    - type of history record
        s_name      - unique name of the temporal
                      relation schema being stored
        object      - temporal object
        relation    - temporal relationship
        start_event - the name of the event which
                      started the temporal relationship
```

A history end record has the following format:

```
( def_external_data  history_end_rec
        ( rec_type              0   3: char: symbol )
       *( s_name                3  20: char: symbol )
        ( end_event            23  10: char: symbol ))
```

The *rec_type* and *s_name* fields are the same as in the start record. The *s_name* field provides the link between the *history_start* and *history_end* records. The field *end_event* contains the name of the event which terminated the relationship. The history event record is specific to the type of temporal object being stored. The basic event structure is as follows:

```
( def_external_data  history_event_rec
        ( rec_type              0  3  : char : symbol )
        ( s_name                3 20  : char : symbol )
        ( event_type           23 10  : char : symbol )
        ( object               33 15  : char : symbol )
        ( time_str             48 17  : char : string )
        ( DATA_SPECIFIC_ATTRIBUTES ) )
```

The only new fields introduced are *event_type* which indicates the type of event and *time_str* which is the numeric representation of the time of the event. The *DATA_SPECIFIC_ATTRIBUTES* are dependent on the temporal object type. Figure 4-2 is an example of the history records which would be generated for the initiation and termination of a sootblower temporal relationship.

Sootblower History Start Record

str	TP_REL233	IR_13	avail	event_55

Sootblower History Event Record

sbe	event_55	sootblower	IR_13	91-01-20/12:12:12	TP-REL23	YES

Sootblower History End Record

end	TP_REL23	event_100

Figure 4-2 History Record Examples for a Sootblower Temporal Relationship

Within the temporal relation schema there are two attributes, discussed previously, *start_hfile* and *end_hfile* which are used for the purpose of updating temporal history. When a temporal relationship is initiated a *start_hifle* attribute with no value is asserted into the temporal relation schema. The temporal history processor is continually trying to match a temporal relation schema which has a *start_hfile* attribute with a value which is not equal to the current history file schema. When this match occurs the temporal relation schema is written to the current history file and the *start_hfile* attribute is given a value of the current history file. If a new history file is opened this match would occur again because the *start_hfile* would be incorrect as it would contain the value of the old history file. In the same fashion the *end_hfile* attribute is used to record the termination of a temporal relationship.

4.4 Querying the Temporal Databases

The query processor was designed to transparently access either the current or history temporal databases depending on the type of query. A transparent query is one in which the query processor will automatically determine if an access of temporal history is required and will perform the necessary actions accordingly. A simple query mechanism was developed where a valid query type consisted of the temporal object of interest and time for which the temporal relationship query is to be made. A query is initiated by asserting a query schema which has the following attributes:

```
( schema ?query
        ( instance_of          query     )
        ( qry_type             relation  )
        ( qry_request          rule_base )
        ( qry_object           ?object   )
        ( qry_time             ?time     )
        ( qry_complete         No        ) )
        ( history              ?his_sync )
        ( history_file         ?h_file   ) )
```

The *qry_type* and *qry_request* attributes of the query schema specify the type of query. The *qry_object* specifies the temporal object of interest in the query. The *query_time* is the time at which the query was initiated and *qry_complete* specifies if the query has been processed. The *history* and *history_file* attributes are asserted into the query schema once the query processor has determined that the information required is in a history file. To choose which history file to load from disk, the *qry_time* is compared to the history file schema to determine which file contains the relationship in question.

Once the history file is identified and placed in the history file attribute, the value of the *history* attribute is updated to *READ*. The history processor contains a rule which matches on a query whose history attribute contains the value READ. This initiates the reading of the history file. All temporal data loaded from a history file is marked as history via a *history* attribute whose value is the name of the query history file. When the history file data is loaded into the temporal database, the history attribute in the query schema is updated to the value LOADED.

Since the history file is now loaded, the query processor attempts to satisfy the query for the query temporal object. If a match on a temporal relationship is found for the temporal object at the required query time, the temporal relationship is asserted into the query schema. The *history* attribute is then updated to PURGE and the *qry_complete* is updated to yes. The temporal history processor has a rule which matches a query schema with a *history* attribute having the value PURGE. As a result all temporal schemas containing the history attribute whose value is the query history file are retracted. On completion of the purge the query schema is deleted.

This query mechanism is complicated in that events must be sequenced. As well, a query must be implemented in two stages. The first stage is a rule which generates a query. The second stage is a rule which waits for the completion of the query and uses the result for the desired function.

As discussed earlier this is a simple query processor which could be further developed; however, it is adequate for the immediate needs of the SAX system.

5 IMPLEMENTATION OF THE SOOTBLOWING ADVISORY EXPERT

5.1 Introduction

This chapter discusses the implementation of SAX which encompasses all the components covered so far; temporal preprocessor, temporal history processor and the query processor and in addition, contains one of the most important components of SAX, the domain knowledge base. The domain knowledge base consists of the rules which determine the sootblowing strategies by utilizing the above mentioned components. An interactive user interface was developed for SAX which allows the operator to enter information for use by SAX as well as provides the operator with the continuous feedback of recommended sootblowing strategies. Figure 5-1 shows the SAX system with all of its component parts. Table 5-1 and 5-2 summarizes the software developed for SAX.

5.2 SAX Knowledge Acquisition

The knowledge base for SAX was acquired from various sources. They were:

1. Plant Operations
2. Plant Engineers
3. Plant Operators
4. Sootblowing Log Book
5. Operation Log Book
6. Strip Charts
7. Hourly Logs
8. Shift Logs
9. Online Performance Monitoring System

Knowledge Acquisition was conducted in five steps presented as follows:

1. Familiarization of the problem
2. Interviews
3. Data Gathering
4. Formulation of the knowledge base rules
5. Verification of the knowledge base rules

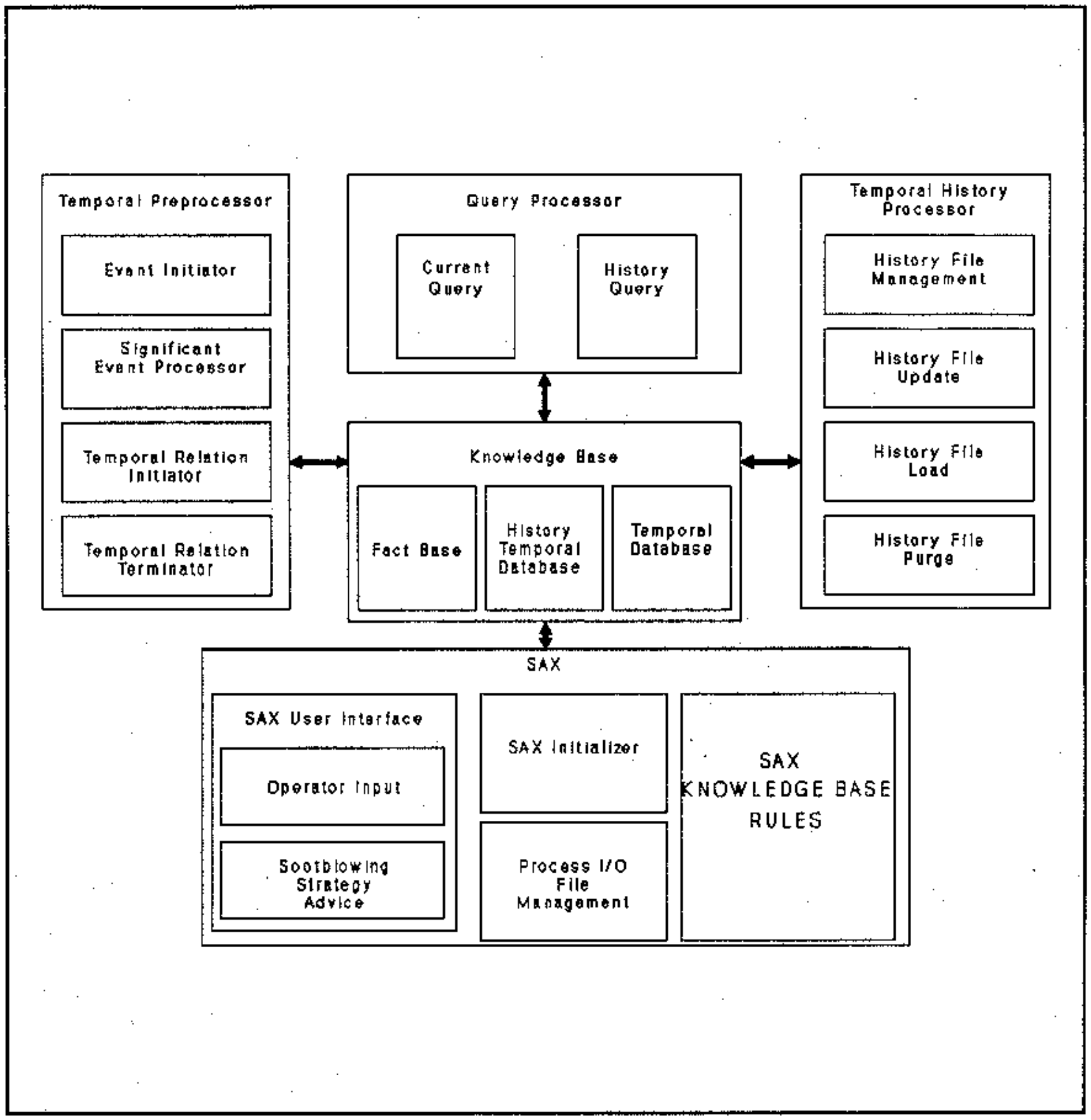


Figure 5-1 SAX System

Familiarization of the problem was attained from research carried out at the Department of Chemical Engineering, University of Waterloo. This research involved empirical studies of the mechanisms of fouling on the walls of large boiler combustion chambers. Studies of three different power stations burning different types of fuel provided a foundation for understanding fouling characteristics associated with fuel types [WYNN79][WYNN80]. Along with this, papers on steam temperature control and general boiler control provided a basic understanding of the control concepts and in particular steam temperature control. [MARI90] [DUKE86].

ART-IM Schema Classes	8
ART-IM Schemas Runtime	1000
ART-IM Rules	140
ART-IM Functions	85
Turbo C Functions	8

Table 5-1 Summary of SAX Code

ART-IM Lines of Code	
Functions	1233
Schemas	3049
Rules	7824
Turbo C Lines of Code	
Functions	505
Total Lines of Code	12611

Table 5-2 Summary of SAX Lines of Code

Once an understanding of the problem was achieved, interviews were conducted. The first interview included plant engineers from Dalhousie Unit #2. Appendix I contains the set of questions presented and information gathered.

The second set of interviews was conducted with plant operations with particular emphasis on the Online Performance Monitoring System. From this interview it was determined that the monitoring system was inadequate and the data calculated for the purposes of slag control and heat absorption was unreliable. This was later confirmed by the operators.

A third set of interviews involved talking with operators during plant operation. This was an informal setting where observations of operator action, based on plant state, was the most important information gathered. From these interviews which included different operators but similar operating conditions, different approaches to sootblowing were established. It became clear that in most cases there were several solutions to a particular sootblowing situation.

In order to obtain enough information about sootblowing during plant operation the operators were given a ***sootblowing information form*** found in Appendix II, which they were to fill out at each sootblowing operation, for a period of two weeks. The information included process variables before and after the sootblowing operation, the sootblowers operated and the reason for the sootblowing operation. For the same two week period the strip charts, hourly logs, and shift logs were obtained. This information was summarized and recorded in Appendix III.

From this information a set of sootblowing rules was defined (see Appendix IV). These rules were presented to the operators at Dalhousie for verification. The necessary modifications were made. Further verification of the rules are discussed in Chapter 6 during the testing phase of SAX.

The knowledge acquisition phase in this thesis proved to be very difficult. There appeared to be a broad consensus in the area of steam temperature control but when it came to when, why and how to operate sootblowers many different approaches were used. Data obtained about mills in operation and fuel quality play an important part in predicting how slag is formed and deposited on the heat transfer surfaces of the boiler. At Dalhousie Unit #2, data in this area was not available to support developing an extensive set of rules for sootblowing. As well, after this thesis was under way, the main source for obtaining heat absorption information, the Online Performance

Monitoring System, was removed from operation. As a result heat absorption process I/O was not incorporated in the SAX system. The availability of this information was important in the detection and confirmation of slag buildup. However, this was not the only means of detecting slag buildup rather it would have provided an extra degree of certainty in the detection process.

5.3 SAX User Interface

The SAX user interface provides the operator with an interactive environment in which he can enter plant information and at the same time obtain sootblowing strategy advice. Selection of options is via a mouse interface.

The SAX user interface was designed based on the standard user interface toolkit provided in ART-IM. This toolkit provided the means to construct standard interfaces consisting of the following:

1. Status line
2. Menu bar with pull-down menus
3. Scrollable, overlapping windows
4. Specialized windows called Dialog Boxes for displaying messages and prompting for answers or data.
5. Input handling via a keyboard or mouse
6. Hypertext Help System
7. Sound Generation

The second important feature of the UI toolkit is its capability to accept asynchronous input while SAX is still executing. This is essential in a real-time system since process I/O must be acquired periodically independent of operator actions in the user interface.

5.3.1 **Main Menu**

The SAX main menu is a menu bar containing pull-down menus. Figure 5-2 shows the SAX Main Menu. This menu provides the means to select one of four items.

```
1.    Sootblowing Operations
2.    Boiler Observations
3.    Fuel/Mill Entry
4.    Exit SAX
```

5.3.2 **Sootblowing Operations**

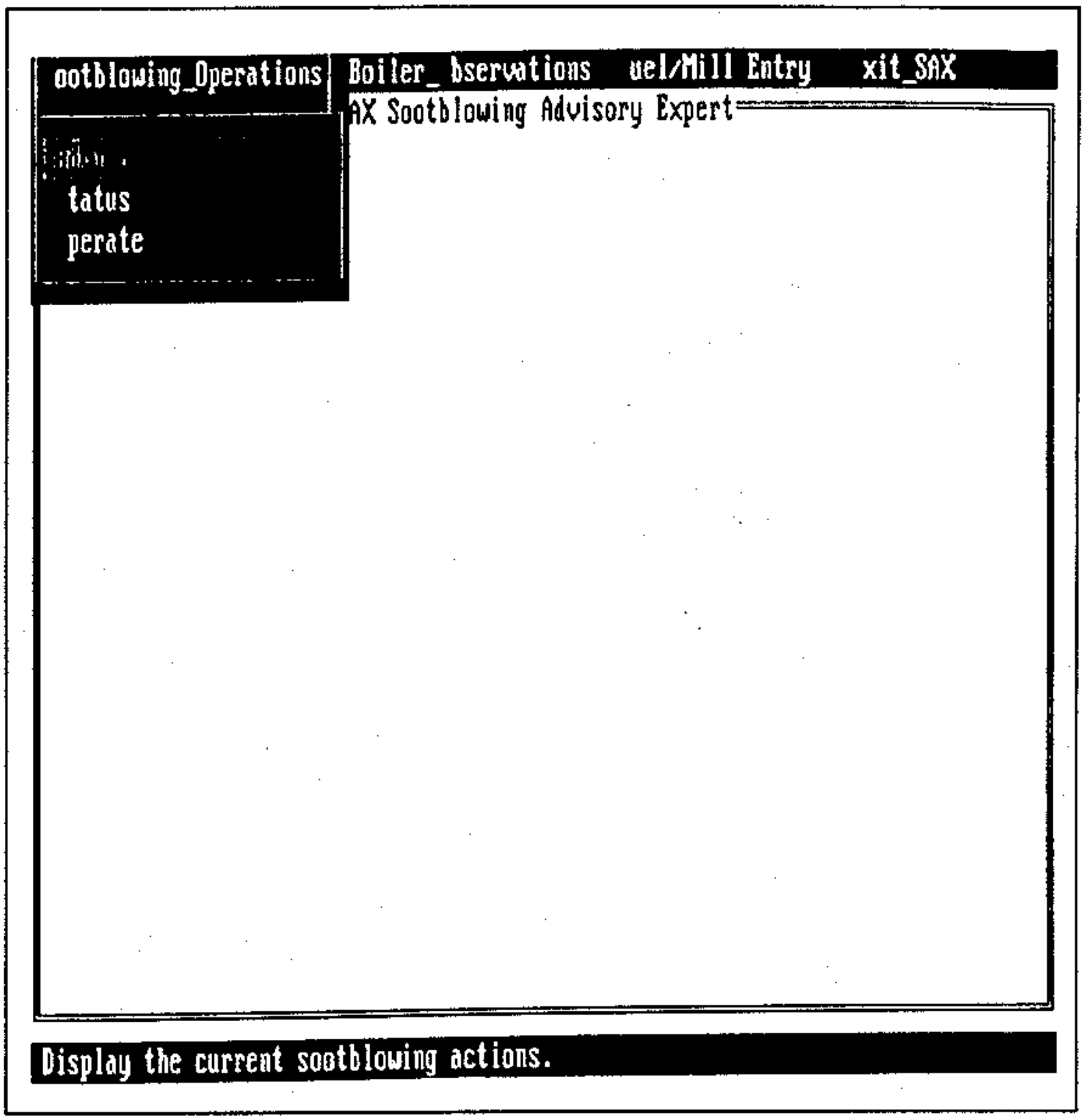
The Sootblowing Operations pull-down menu consists of three components:

```
1.    Advice
2.    Status
3.    Operate
```

Advice

The sootblowing advice is the key component of the user interface and presents the sootblowing strategies to the operator. The basic design consists of a sootblower action list and a sootblower advice window. The sootblower action list as shown in Figure 5-3, is a scrollable list where each item in the list is a sootblowing strategy recommended by SAX. A sootblowing strategy can consist of one or more sootblowers which should be operated together. For each sootblowing strategy in the list it is possible to delete or obtain an explanation of why the strategy was chosen by SAX.

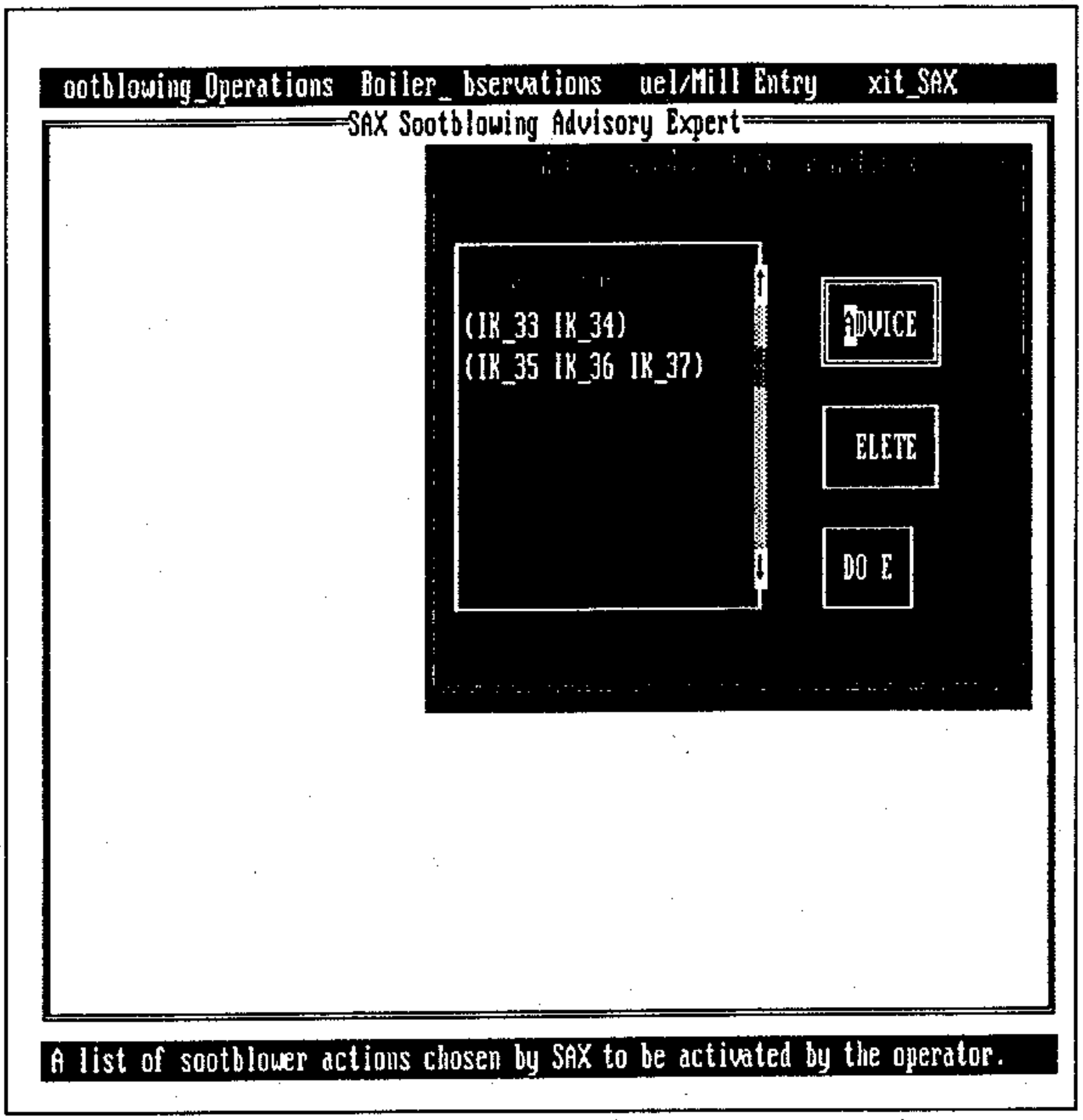
The *delete* action removes the strategy from the list. It was provided to allow deletion of a strategy which the operator felt was incorrect. If the condition which originally placed the strategy on the list is still in existence, the strategy will again be placed on the list. This type of action can be used to detect errors or refinements required in the SAX knowledge base.



Display the current sootblowing actions.

Figure 5-2 SAX Main Menu/Window

The second option provided is *advice*. When invoked for a selected sootblowing strategy an advice window pops up as shown in Figure 5-4. In this window the sootblowing strategy, the date the strategy was invoked, and the reason the strategy was chosen by SAX are displayed.


ootblowing_Operations Boiler_ bservations uel/Mill Entry xit_SAX
SAX Sootblowing Advisory Expert
DVICE
ELETE
DO E

(IK_33 IK_34)
(IK_35 IK_36 IK_37)

A list of sootblower actions chosen by SAX to be activated by the operator.

Figure 5-3 Sootblowing Strategy List

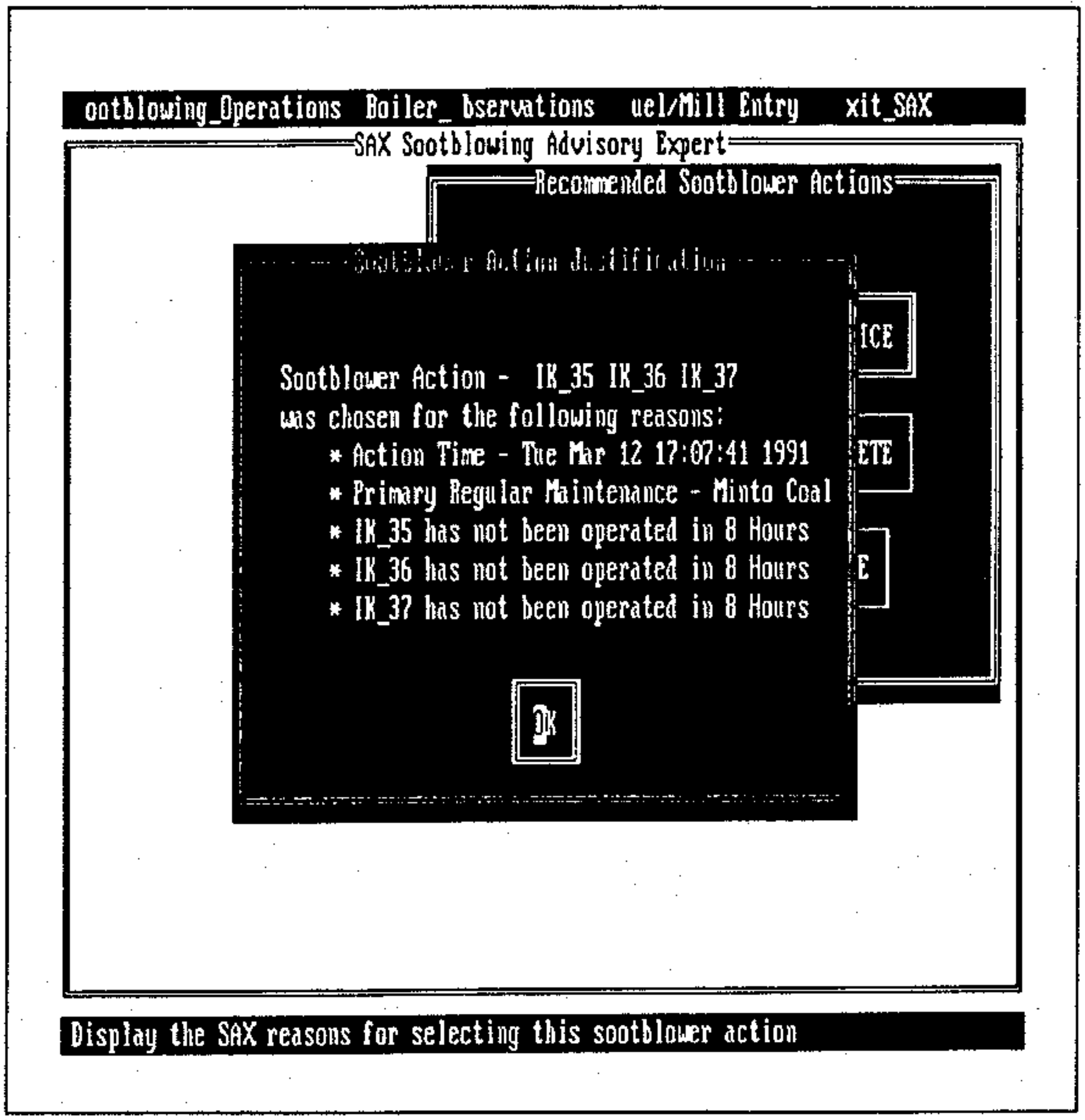
The original intent for the sootblowing advice component was to use the ARTS-IM's justification system which tracked how and why facts and schemas were placed in the knowledge base. However, this was not possible since justification data for items retracted from the knowledge base became unavailable for use. Since SAX is a real-time system, its data is constantly changing making this problem more prevalent. It was decided that a simple tracking method could be used to determine the reasoning behind a strategy choice. This will be discussed later in this chapter.

Status

The status option provides the means to obtain the current status of a particular sootblower. This is done by successively moving down the sootblower schema hierarchy, for example, from a plant sootblower to a waterwall sootblower to a section of the waterwall, and finally to the list of sootblowers of which one can be chosen. This interface was accomplished using a series of multiple choice dialog boxes. The status information displayed includes the sootblower name, the date last operated, whether it is available to be operated and whether it is currently in the sootblower action list. This is illustrated in Figure 5-5. A deficiency in the user interface design is found in the limitation of selecting one sootblower at a time. For instance, if an operator required the status of four sootblowers in the lower waterwall he/she would need to repeat the entire sootblowing selection process four times. A more efficient approach would be to travel the hierarchy to the desired area and select all sootblowers required for status. With the toolkit provided there was no readily available UI tool to provide multiple selections in a dialog box. For the purposes of the SAX prototype, it was not important to attain the most efficient user interface. If SAX were to be implemented in the plant environment, however, the user interface would be unacceptable as it contains too many steps making it time consuming and difficult to use.

Operate

Since SAX is an open loop system, once a sootblowing strategy is proposed, and sootblowers are operated (independent of the SAX system), it is up to the operator to update the SAX system with the sootblower operation information. The selection of the sootblower is achieved in the same way as explained above for sootblower status.

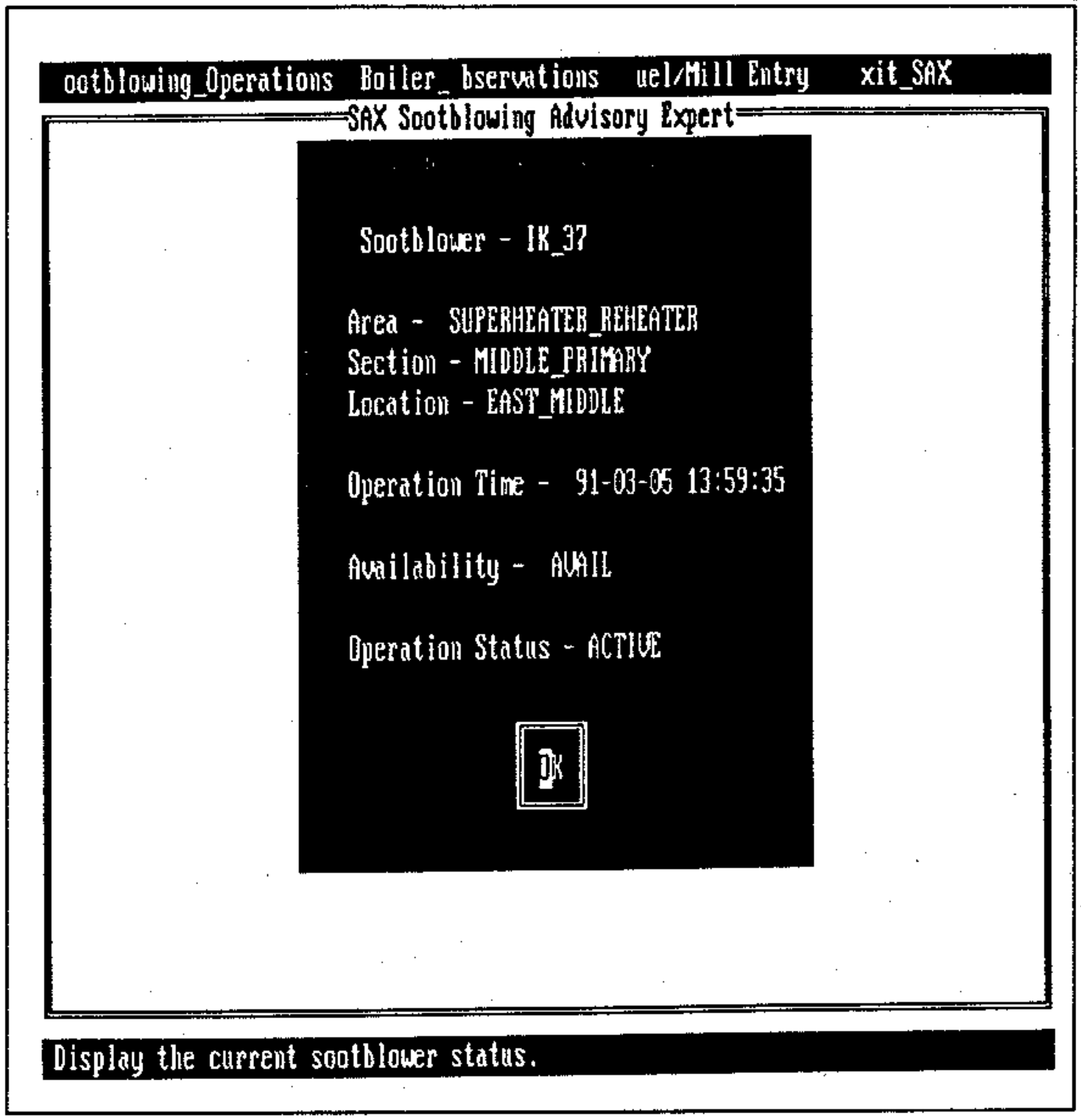

ootblowing_Operations Boiler_ bservations uel/Mill Entry xit_SAX
SAX Sootblowing Advisory Expert
Recommended Sootblower Actions
Sootblower Action Justification
ICE
ETE
E
OK

```
Sootblower Action -  IK_35 IK_36 IK_37
was chosen for the following reasons:
    * Action Time - Tue Mar 12 17:07:41 1991
    * Primary Regular Maintenance - Minto Coal
    * IK_35 has not been operated in 8 Hours
    * IK_36 has not been operated in 8 Hours
    * IK_37 has not been operated in 8 Hours
```

```
Display the SAX reasons for selecting this sootblower action
```

Figure 5-4 Sootblowing Advice Window

Once the selection has occurred an input window pops up to request the time at which the sootblower was operated. This provides the time stamp for the sootblower temporal event. An acknowledgement of a sootblower operation is shown in the message window in Figure 5-6.


ootblowing_Operations Boiler_ bservations uel/Mill Entry xit_SAX
SAX Sootblowing Advisory Expert
OK

Sootblower - IK_37

Area - SUPERHEATER_REHEATER
Section - MIDDLE_PRIMARY
Location - EAST_MIDDLE

Operation Time - 91-03-05 13:59:35

Availability - AVAIL

Operation Status - ACTIVE

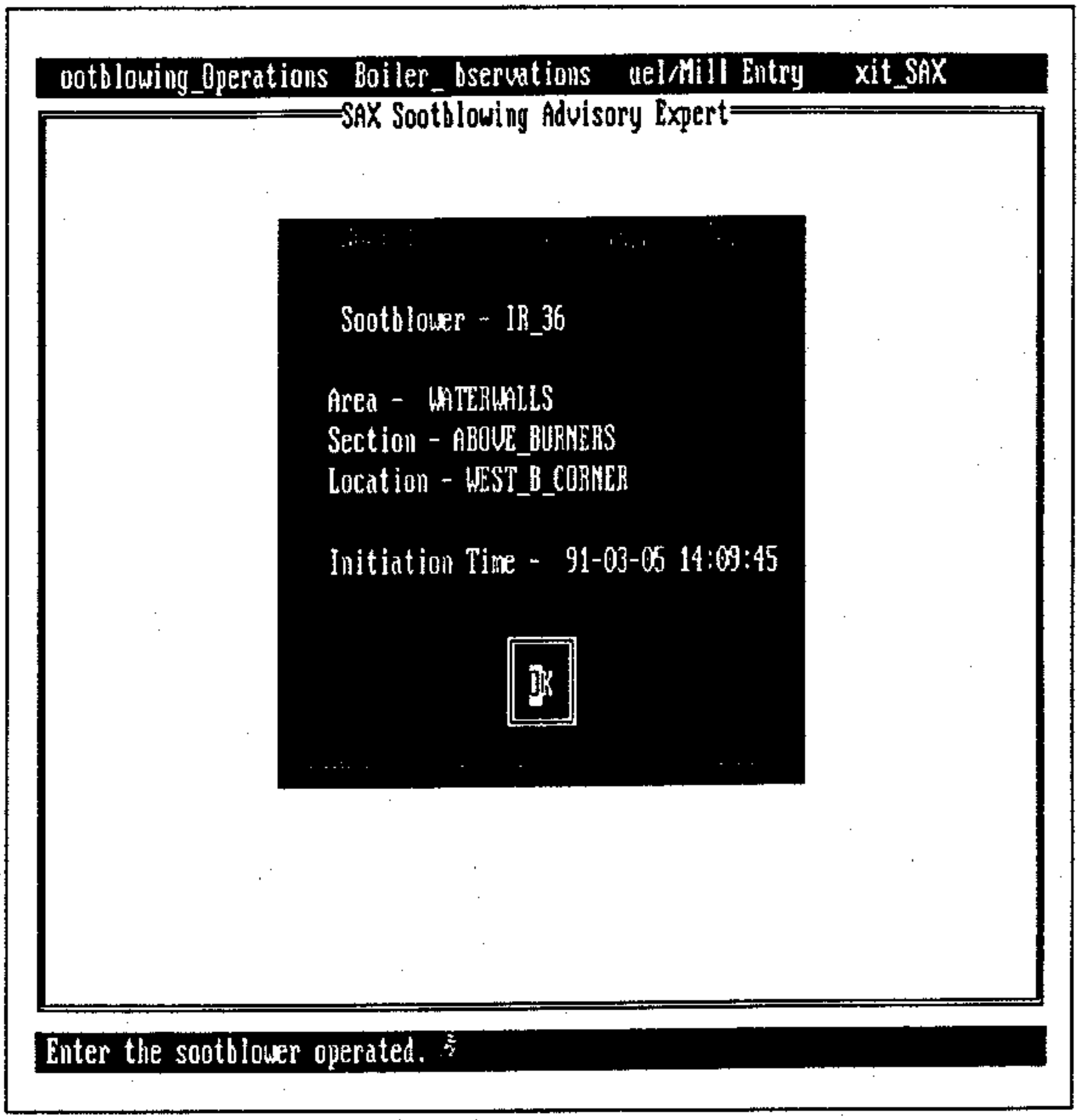
Display the current sootblower status.

Figure 5-5 Sootblower Status Window

5.3.3 **Boiler Observations**

The boiler observation pull-down menu consists of the following options:

```
1.  Status
2.  Clear
3.  Change
```


ootblowing_Operations Boiler_ bservations uel/Mill Entry xit_SAX
SAX Sootblowing Advisory Expert
OK

Sootblower - IR_36

Area - WATERWALLS
Section - ABOVE_BURNERS
Location - WEST_B_CORNER

Initiation Time - 91-03-05 14:09:45

Enter the sootblower operated.

Figure 5-6 Sootblower Operation Window

Status

During every shift at the plant, observations are made to determine the slagging conditions in the boiler. These observations access the buildup of slag or dust on the surfaces of the boiler and are used by the SAX system to select or support proposed sootblowing strategies.

In order to provide a user interface which the operator could use effectively to enter the boiler observations, it was necessary to understand where and how these observations were recorded. From the sootblowing log book it was determined that the boiler could be divided into areas of observation. Each area could then be divided based on orientation, that is the north, east, south and west walls as shown in Figure 5-7. At the lowest level each wall can be divided into the left corner, middle, and right corner. With this organization the following areas were defined:

1. Slopes
2. Lower waterwall
3. Middle waterwall
4. Upper Waterwall
5. Secondary
6. Cold Reheat
7. Hot Reheat
8. Final Reheat
9. Primary

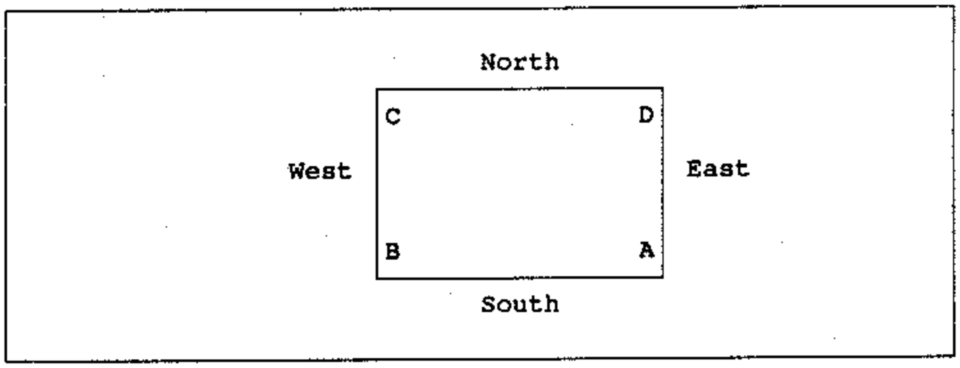


Figure 5-7 Orientation of Boiler Walls

It was deemed important that the operator should be able to, at a glance, obtain a general idea of the observed state of the boiler. To provide this a spreadsheet type of interface was designed for boiler observations as shown in Figure 5-8. The columns of the spreadsheet represent the boiler areas and the rows of the spreadsheet represent the orientation of each area. The actual entry in the spreadsheet indicates the current temporal relationship for this area.

Clear and Modify

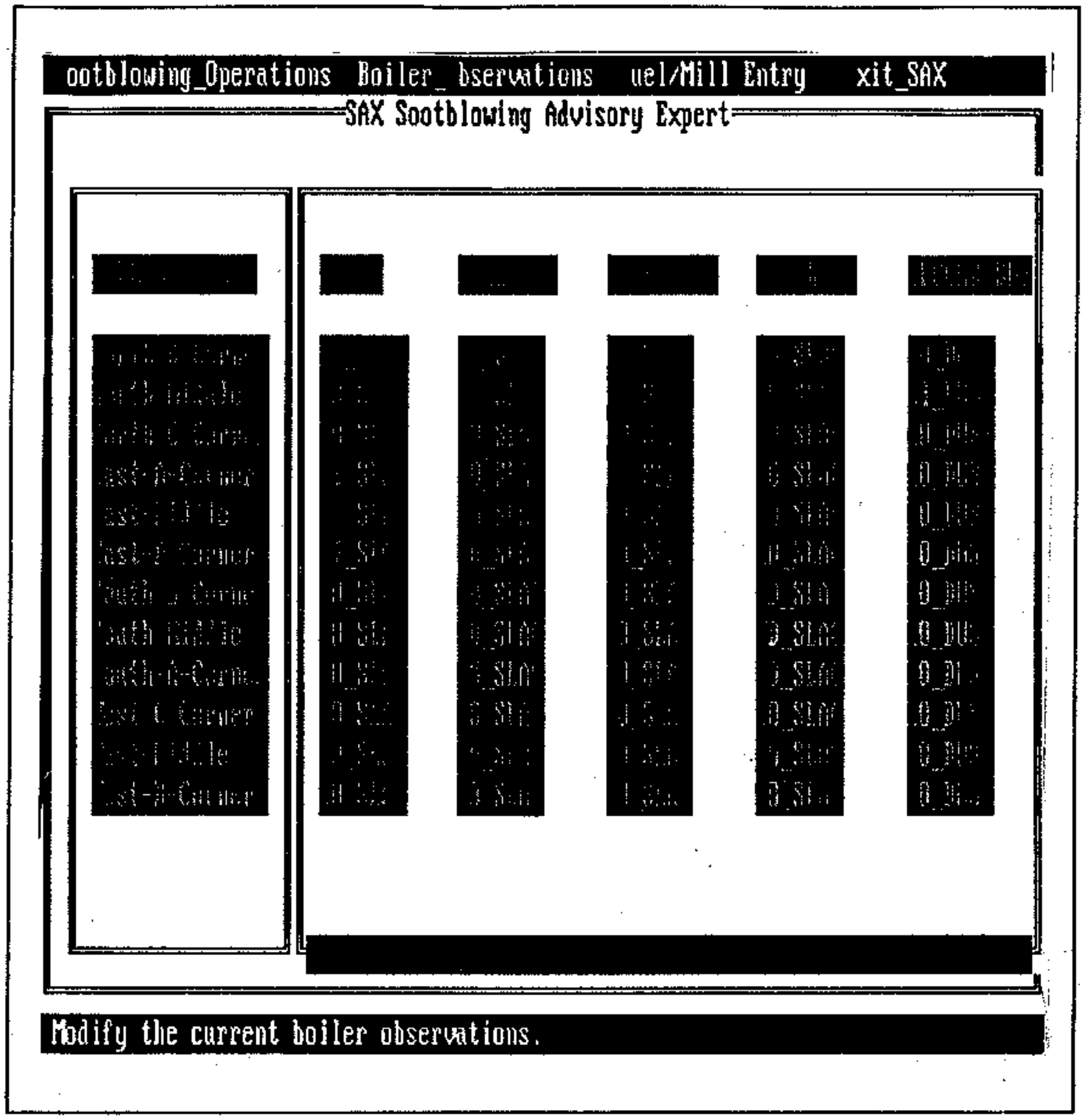
To modify an observation two functions were provided, a clear function and a modify function. The clear function provided a method to change all observations to a no_slag state or in other words to a clean boiler state.

The modify option provided a method to selectively enter boiler observations. Entry of boiler observations can be made on three levels: by area, (ie. upper waterwall), by wall (ie. north) or by wall component (ie. corner or middle).

When a selection is made the operator is prompted for the slag type, the amount in inches and the time the observation was made. Once these inputs are obtained the temporal preprocessor initiates the appropriate temporal relationships.

5.3.4 **Fuel/Mill Entry**

The fuel/mill section of the user interface allows the operator to enter information and obtain the status on the fuel being burned and the mills currently operating in the plant. When the fuel/mill option is chosen on the main menu the pull-down menu is displayed and contains the following options:


ootblowing_Operations Boiler_ bservations uel/Mill Entry xit_SAX
SAX Sootblowing Advisory Expert

Modify the current boiler observations.

Figure 5-8 Boiler Observation Window

1. Status Fuel/Mill
2. Clear Fuel/Mill Status
3. Enter Fuel Type
4. Enter Mills Operating

Status Fuel/Mill

The status option provides information about the fuel being burned and the mills currently operating. The status window is shown in Figure 5-9 and contains the fuel type being utilized in the plant. Associated with the fuel type are two distinguishing characteristics. They are the percentage of moisture in the fuel and the BTU content of the fuel. These values are obtained from tests performed on the fuel by operating staff at the plant. These two characteristics provide a measure of the quality and burning characteristics of the coal. Along with fuel information the status window provides a list of the currently operating mills, where a mill is the mechanical equipment through which the coal is crushed, passes and is ignited.

Select Fuel Type

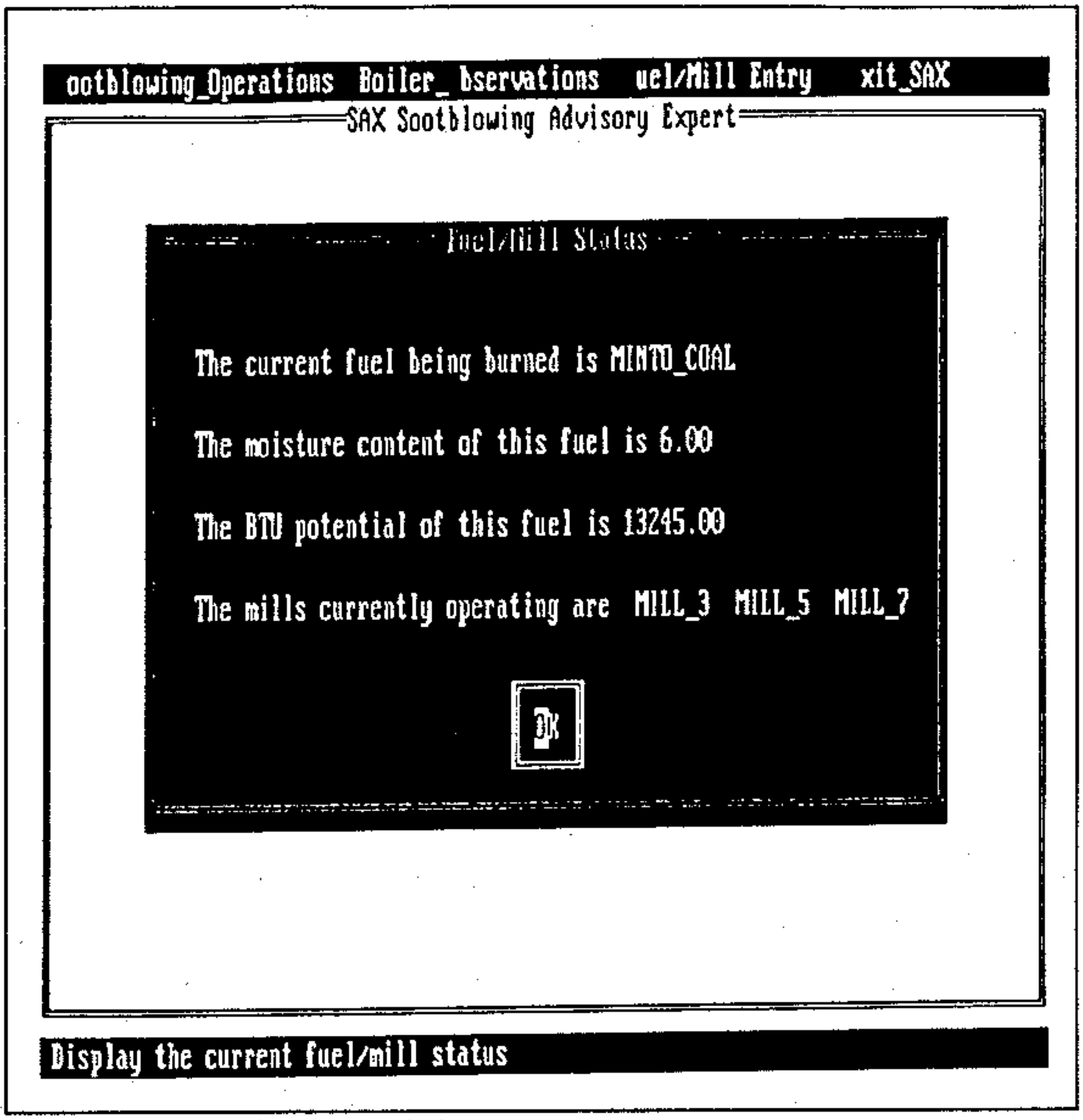
From recent history in the Dalhousie Plant a list of possible fuel types was defined as follows:

```
1.    Minto Coal
2.    Lingan/Phalen
3.    Columbian-Maturin Coal
4.    West Virginian Farrel
```

When the fuel type is changed in the plant, the operator must inform the SAX system of this change. This is accomplished by selecting Select Fuel Type in the Fuel/Mill Entry pulldown menu. From this the list of fuel types is presented in a multiple choice window (Figure 5-10). The operator can select one of the fuels. The operator is then provided with input windows to enter the moisture content of the fuel and the BTU content of the fuel type in question. The operator must then enter the time the fuel was changed in the plant.

Enter Mills Operating

When the mills operating in the plant are changed the operator must inform the SAX system by selecting the mill option in the Fuel/Mill Entry pulldown menu. A checkbox type window is displayed, shown in Figure 5-11, which allows the selection


ootblowing_Operations Boiler_ bservations uel/Mill Entry xit_SAX
SAX Sootblowing Advisory Expert
Fuel/Mill Status
OK

The current fuel being burned is MINTO_COAL

The moisture content of this fuel is 6.00

The BTU potential of this fuel is 13245.00

The mills currently operating are MILL_3 MILL_5 MILL_7

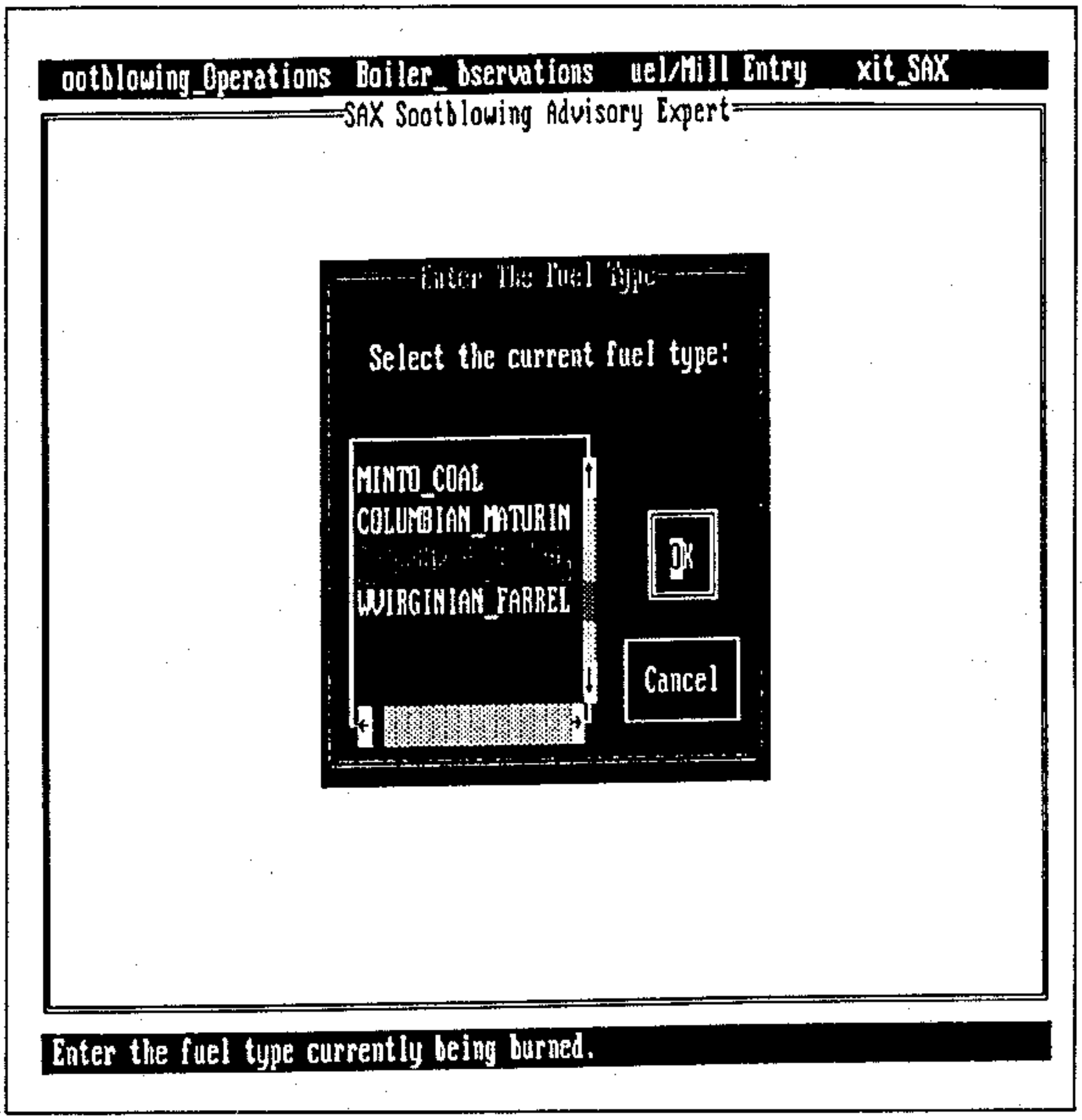
Display the current fuel/mill status

Figure 5-9 Fuel/Mill Status Window

of one or more mills by the operator. The time the mill configuration changed must be entered to provide a time stamp for the temporal event in question.

5.3.5 Exit SAX

The final item on the bar menu is Exit SAX. Selection of this item provides the means to terminate the running of the SAX System.


ootblowing_Operations Boiler_ bservations uel/Mill Entry xit_SAX
SAX Sootblowing Advisory Expert
Enter The Fuel Type
OK
Cancel

Select the current fuel type:

MINTO_COAL
COLUMBIAN_MATURIN
WVIRGINIAN_FARREL

Enter the fuel type currently being burned.

Figure 5-10 Fuel Selection Window

5.4 Knowledge Base

The components of SAX which have been discussed so far in this report were created as tools to be utilized by the SAX knowledge base. The temporal preprocessor and the history processor both maintain temporal information which is used by the SAX knowledge base to make sootblowing strategy decisions.

The SAX knowledge base can be divided into three sections. The first section, periodic sootblowing strategies, invokes sootblowing strategies based on periodic

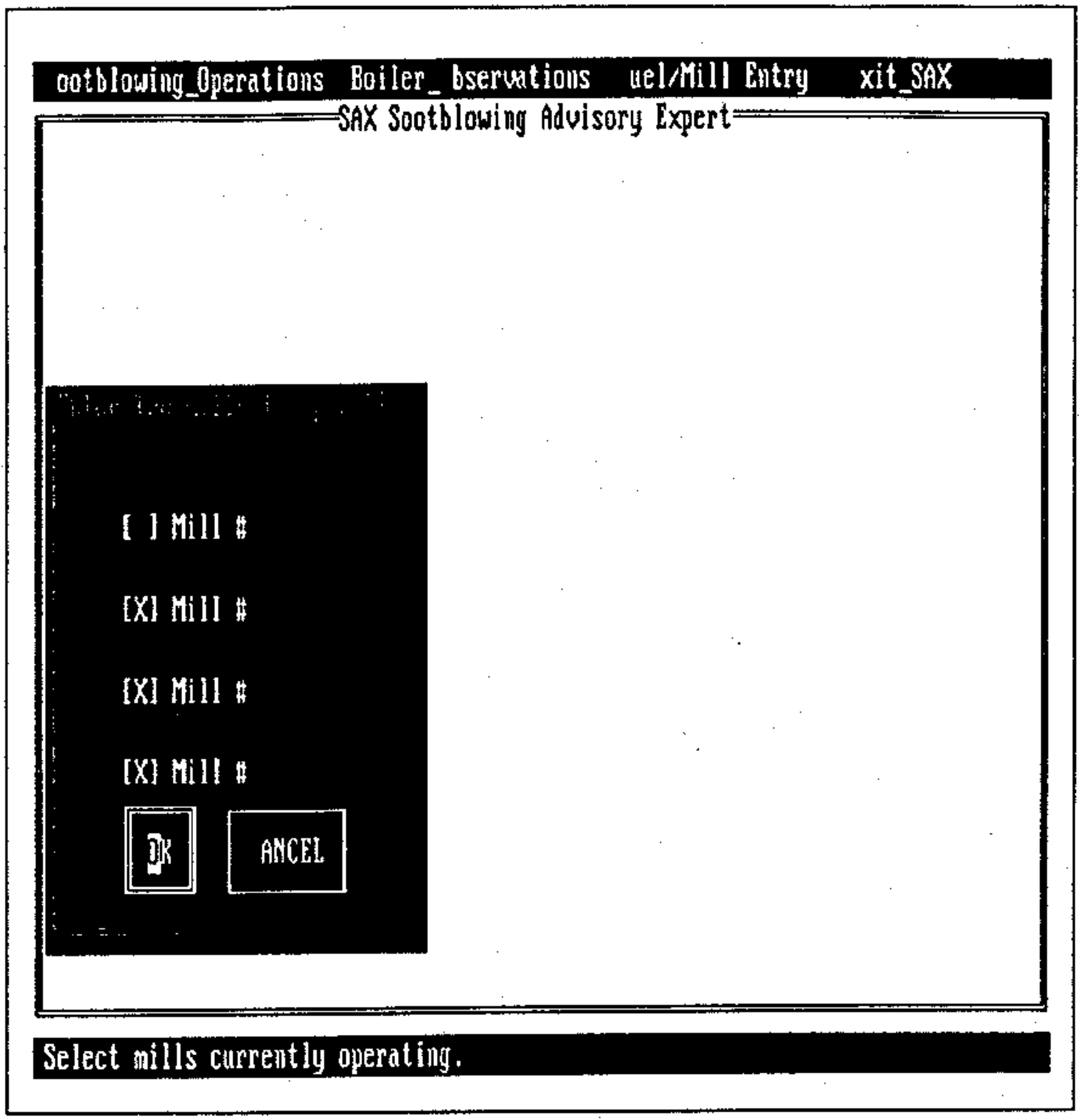

ootblowing_Operations Boiler_ bservations uel/Mill Entry xit_SAX
SAX Sootblowing Advisory Expert
[ ] Mill #
[X] Mill #
[X] Mill #
[X] Mill #
OK
ANCEL
Select mills currently operating.

Figure 5-11 Mill Operation Window

sootblowing rules established by plant operating staff. The second section, observation sootblowing strategies, initiates sootblowing strategies based on boiler observations entered by the operator via the SAX user interface. The last section, plant state sootblowing strategies, invokes sootblowing strategies based on the established state of the plant. Once a requirement for a sootblowing strategy is established, a selection from a defined set of sootblowing strategies is made. How this is accomplished will be discussed later in this chapter.

5.4.1 **Periodic Sootblowing Strategies**

Plant operations have established periodic sootblowing rules for various areas of the boiler. These periodic rules are based on the fuel type and mill configuration. For example, the following rules for the primary, economizer and air heater areas of the boiler were established for the fuel Minto Coal:

1. In the primary region, operate 3 sootblowers during every shift.
2. In the air heater region, every shift, operate IKAH_1 and IKAH_2 sootblowers. On alternate shifts operate IKAH_3 and IKAH_4 sootblowers.
3. In the economizer region operate all sootblowers, IK_33 and IK_34 every 4 weeks.

These periodic rules were developed using a timing mechanism. When a periodic sootblower timer expired an appropriate sootblowing strategy was chosen and placed on the sootblowing strategy list.

5.4.2 **Observation Sootblowing Strategies**

Rules have been established by plant operations with regards to sootblowing requirements based on the boiler observations which are periodically made by the plant operators. The standard rule is as follows:

1. If a medium to high buildup of slag or dust is observed by the operator during his periodic boiler observation task then choose appropriate sootblowers to remove the buildup.

This rule is implemented in SAX using the boiler observation temporal relationships. If a boiler section has a temporal relationship of dust_med, dust_high, slag_medium or slag_high then an appropriate sootblowing strategy must be chosen. This strategy is chosen based on the sootblowers which are defined for that area of the boiler. This link between boiler observation sections and individual sootblowers was established during definition of the schema hierarchies. In a sootblower schema an attribute, *orientation* is defined. The value of this attribute is the boiler section to which this sootblower is associated and if operated, would remove the slag or dust buildup. An example of this would be as follows:

```
Sootblower:
        ( defschema   IR_1
                ( instance_of lower_waterwall_sootblower )
                ( orientation  lww_s_bcor )
                ( location south_b_corner ))

boiler section:
        ( defschema   lww_s_bcor
                ( instance_of  lww_south )
                ( row_pos 9 ))
```

Observation rules collect all the sootblowers for a boiler section and then asserts a sootblowing strategy schema which is added to the sootblower strategy list.

5.4.3 Plant State Sootblowing Strategies

These rules were developed from known operating knowledge and heuristic information obtained from plant operations combined with the evaluation of plant operating data.

These rules are established based on combinations of variable temporal relationships which, together, infer a plant state indicative of slag or dust buildup. The following sample rules determine which sootblowing actions are required based on a series of plant states. Appendix III contains a complete list of the plant state sootblowing rules.

PLANT_STATE_3

```
IF the plant is operating at full load AND
   the control_system_stable_timer is zero AND
   this plant state has been in existence for a
   minimum of time determined by the
   PLANT_STATE_3 timer AND
   Reheat_Steam_Temp is in the high range AND
   Main_Steam_Temp is in the norm range AND
   Gas_Exit_Temp is in the high range AND
   Burner_Tilt is in the low range AND
   Excess_Air is in the normal range AND
   Reheat_Steam_Spray is in the normal range AND
   Main_Steam_Spray is in the normal range
THEN
   assert slag_primary_section AND
   assert slag_upper_waterwall_section AND
   assert slag_middle_waterwall_section
```

PLANT_STATE_4

```
IF the plant is operating at full load AND
   the control_system_stable_timer is zero AND
   this plant state has been in existence for a
   minimum of time determined by the
   PLANT_STATE_4 timer AND
   Reheat_Steam_Temp is in the normal range AND
   Main_Steam_Temp is in the high range AND
   Gas_Exit_Temp is in the high range AND
   Burner_Tilt is in the low range AND
   Excess_Air is in the normal range AND
   Reheat_Steam_Spray is in the normal range AND
   Main_Steam_Spray is in the normal range
THEN
   assert slag_middle_waterwall_section AND
   assert slag_lower_waterwall_section
```

PLANT_STATE_9

```
IF the plant is operating at full load AND
   the control_system_stable_timer is zero AND
   this plant state has been in existence for a
   minimum of time determined by the
   PLANT_STATE_9 timer AND
   Reheat_Steam_Temp is in the low range AND
   Main_Steam_Temp is in the normal range AND
   Gas_Exit_Temp is in the high range AND
   Burner_Tilt is in the low range AND
   Excess_Air is in the normal range AND
   Reheat_Steam_Spray is in the normal range AND
   Main_Steam_Spray is in the normal range
THEN
   assert slag_hot_reheat_section
   assert slag_secondary_section
```

PLANT_STATE_13

```
IF the plant is operating at full load AND
   the control_system_stable_timer is zero AND
   this plant state has been in existence for a
   minimum of time determined by the
   PLANT_STATE_13 timer AND
   Reheat_Steam_Temp is in the high range AND
   Main_Steam_Temp is in the high range AND
   Gas_Exit_Temp is in the high range AND
   Burner_Tilt is in the low range AND
   Excess_Air is in the low range AND
   Reheat_Steam_Spray is in the high range AND
   Main_Steam_Spray is in the high range
THEN
   assert slag_upper_waterwall_section AND
   assert slag_middle_waterwall_section AND
```

```
PLANT_STATE_17

IF the plant is operating at full load AND
   the control_system_stable_timer is zero AND
   this plant state has been in existence for a
   minimum of time determined by the
   PLANT_STATE_17 timer AND
   Reheat_Steam_Temp is in the normal range AND
   Main_Steam_Temp is in the high range AND
   Gas_Exit_Temp is in the high range AND
   Burner_Tilt is in the low range AND
   Excess_Air is in the normal range AND
   Reheat_Steam_Spray is in the normal range AND
   Main_Steam_Spray is in the high range
THEN
   assert slag_middle_waterwall_section AND
   assert slag_lower_waterwall_section
```

From these rules, the area of slag buildup is established. Sootblowing strategies are then chosen for the targeted areas.

5.4.4 Selection of Sootblowing Strategies

The selection of sootblowing strategies is based first on establishing the area which requires sootblowing and secondly choosing a sootblower sequence from an established set of sootblowing strategies.

Sootblowing regions were defined. For each of these regions one or more sootblowing strategies were established. A sootblowing strategy consisted of one or more sequences of sootblowers. A sootblower sequence included one or more individual sootblowers. An example of a sootblower strategy as defined by the hierarchial schema structure used is found in Figure 5-12.

```
( defschema primary_strategy
        ( is-a sootblowing_strategy )
        ( sequences ( pri_strat_1 pri_strat_2 pri_strat_3 nil ))
        ( current strategy pri_strat_1 ))

( defschema pri_strat_1
        ( is-a primary_strategy )
        ( strategy 1 )
        ( type staggered_across )
        ( sequences ( ps1_seq1 ps1_seq2 ps1_seq3 nil nil nil ))
        ( current_seq ps1_seq1 )
        ( inoperable_advice "All sootblowers in this strategy
          inoperable" ))

( defschema ps1_seq1
        ( instance-of pri_strat_1 )
        ( seq ( IK_38 IK_39 IK_42 nil )))

( defschema ps1_seq2
        ( instance-of pri_strat_1 )
        ( seq ( IK_32 IK_40 IK_41 nil )))

( defschema ps1_seq3
        ( instance-of pri_strat_1 )
        ( seq ( IK_36 IK_37 IK_42 nil )))
```

Figure 5-12 Example of Primary Sootblowing Strategies

In this example there are three strategies defined for the primary boiler region from a maximum of four possible strategies. They are called *pri_strat_1, pri_strat_2* and *pri_strat_3. Pri_strat_1* is defined as a staggered-across type strategy and contains the sequences of *ps1_seq1, ps1_seq2* and *ps1_seq3.* There are a maximum of six possible sequences which can be defined in each strategy. A sequence within a strategy can have a maximum of four sootblowers as shown in sequence *ps1_seq1.*

For each region, the types of strategies and the combination of sequences defined were dictated by sootblowing patterns established in the day-to-day sootblowing operations in the plant. In the initial formulation of sootblowing strategies an attempt was made to develop sootblowing strategies based on the sootblowers operated during different plant situations. From further analysis of the data it was determined that the same sootblowing strategies were not uniformly used for the same situations.

Selection of sootblowers for operation appeared to be very dependent on the operator and not on the plant situation. For this reason a more generalized approach to the formulation of sootblowing strategies was taken. The following criteria was established based on data collected and sootblowing logs:

1. Divide the boiler into sootblowing areas:

    a) Primary region
    b) Final Reheat region
    c) Hot Reheat region
    d) Cold Reheat region
    e) Secondary Superheater region
    f) Upper Waterwall region
    g) Middle Waterwall region
    h) Lower Waterwall region
    i) Economizer maintenance
    j) Air heater maintenance
    k) Primary maintenance

2. In each area, establish which sootblowers are operated (in some areas there are sootblowers which appear never to be operated).

3. In each area determine if any patterns exist in the grouping of sootblowers operated. Utilize these as sootblowing sequences.

4. Determine if groups of sootblowers are activated in any particular way.

5. If no patterns exist in sootblowing operations sequentially group sootblowers in sequences which can then be utilized sequentially.

In the primary region of the boiler a fairly clear set of sootblower strategies were utilized by operators. They can be classified as staggered_across, double_across, double_one which are named as such for the orientation of the sootblower on the boiler wall as shown in Figure 5-13.

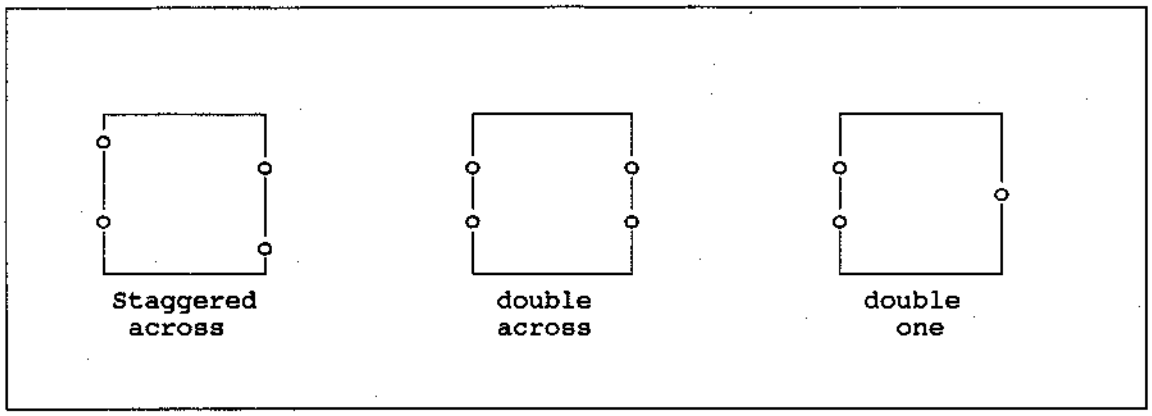


Figure 5-13 Sootblower Strategies Sootblower Orientation

In the reheat and secondary superheater regions, establishing sootblowing strategies proved difficult. From the sootblowing operation data it was determined that the sootblowers in this region were not operated very often and some not at all. It was determined, however, that sootblowers in the final reheat section of the boiler were operated more frequently than either the hot or cold reheat sections. The sootblower strategies in the reheat region were developed in much the same way as the primary region. The secondary superheater region received very little sootblowing action and as a result sootblowers were grouped into sequences of two and placed in a single strategy.

Waterwall sootblowing provided several strategy types as established above for the primary region of the boiler as well as several others due to the fact sootblowers in this region are distributed on all four boiler walls. Some exceptions were made with regards to the use of upper waterwall and lower waterwall sootblowers. These sootblowers appear less frequently in sequences than middle waterwall sootblowers since this was a pattern determined from the plant operation data.

Slope sootblowing was based on boiler observations only. Selection of sootblowing strategies for boiler observations was discussed earlier in this chapter.

Once the sootblowing strategies were determined, utilization of these strategies by SAX occurred sequentially. In this fashion the first strategy for a region was selected. If a sootblowing operation for the region was determined by the SAX knowledge base rules, then the first sequence of the strategy was chosen and placed on the sootblowing strategy list. Each time a sootblowing operation was required the next sequence in the strategy was selected until all sequences in the strategy were utilized. At this point, the next strategy in the region was selected. When all the stratgies were exhausted the selection process would begin at the first sootblowing strategy again.

5.5 SAX Operation

In this section a typical scenario in the operation of SAX will be given which will provide a means of logically joining all that has been discussed so far into one cohesive software package.

When SAX is started, it initializes all the temporal schemas and other miscellaneous variables. Asynchronous operation begins. Process variable information is acquired from an external data source and is processed by the temporal preprocessor generating temporal relationships. The temporal history processor automatically begins the collection of temporal information in the temporal history database. The inference engine at the same time utilizes the SAX knowledge base rules to continuously establish the presence of slag in the boiler. Selection of sootblowers occurs if slag buildup is determined. At the same time this is occurring, SAX is processing interactive operator input from the input devices (keyboard or mouse). Modifications to temporal objects, such as operating a sootblower, entry of boiler observations, changing the fuel types, or changing mill configuration are processed by the temporal preprocessor. This data is again used by the SAX knowledge base rules to determine if slagging conditions have occurred. The operator is also continually monitoring, via the user interface, the sootblowing strategy list to find out the sootblowing strategies recommended by SAX. The justification for the selection of these strategies is available via the sootblowing strategy advice window. At any time the operator can obtain, via the user interface, the status of individual sootblowers, boiler observation status or fuel/mill status.

6 TESTING OF THE SAX SYSTEM

6.1 Introduction

To test the SAX system it was necessary to provide continuous real-time input of process I/O variables. In the course of the development of SAX several methods of testing were discussed. The first option was to use the boiler simulator developed by Maritime Nuclear. This was not considered feasible in the time frame of the thesis, since the simulator required further development to meet the needs of SAX. As well, the simulator software required a conversion between Turbo C and Microsoft C.

The second option considered was to provide a process I/O generator which would supply the process I/O variables over their engineering range. This would require a front end processor in Microsoft C which would capture the process I/O from the generator, manipulate it and pass it to the SAX system. This option was not chosen because the process I/O generator hardware was not readily available. Secondly, it would have been difficult to manually manipulate multiple process I/O variables to produce real plant situations.

The third option considered was the generation of process I/O data files which could be read by the SAX system in real-time. The data contained in these files would be obtained from actual plant operation data. This option provided the advantage of having the plant operation data with which to compare the SAX system operation. This option was chosen as it appeared the most feasible.

6.2 The Collection of Raw Test Data

The generation of the data files was accomplished according to the following steps:

1. Obtain raw test data
2. Generate process I/O files from raw test data
3. Design an ART-IM interface to process I/O data files

As discussed in Chapter 5 and found in Appendix III, plant data was collected for sootblowing operations during a period of time in January and February of 1990. This data focused on sootblowing operations performed by the operators. For each sootblowing operation the operator stated a reason for the actions taken. It was decided to collect plant process data for the variables needed in SAX, based at intervals around key sootblowing operations. For example, in Appendix III, there is a sootblowing Action 12, which occurred on Jan 26, 1990 at 21:50. For this action plant process I/O data was collected over the interval 21:30 to 22:30.

The plant process data was collected from strip charts, hourly logs and shift reports. The following data was obtained by painstakingly taking readings at five minute intervals from strip charts obtained from the plant:

1. Reheat Steam Temperature
2. Main Steam Temperature
3. Main Steam Pressure
4. Excess Air
5. Steam Flow

The following data was obtained from logs which were available at one hour intervals.

1. Gas Exit Temperature
2. Main Steam Attemporating Spray
3. Reheat Steam Attemporating Spray

Finally the Burner Tilt was obtained from the shift report which was generated approximately every eight hours. For the data collected from the hourly logs and shift report it was necessary to manually interpolate the data to produce values at five minute intervals as none were available at this resolution.

Real-time process I/O data for SAX was required every five seconds. The raw data collected was not available at this resolution. For this reason a method of linearly interpolating the data between readings at five second intervals was required. A five second interval was chosen to simulate the real-time processing which would be required in the plant. It should be noted that some accuracy in the data is lost, however, not to the point where it would have a detrimental affect on the testing of the SAX system.

When the test data was compared to the plant operating data provided by the operators, discrepancies were found (ie. reheat temperature given by the operator at time x did not have the equivalent value when read from the strip chart). Given this, some of the sootblowing strategies results provided by SAX may differ from the plant operator chosen strategies.

6.3 Generation of Process I/O Data Files

Once the raw data was collected it was entered in Lotus 123 for the purposes of thesis presentation as well as to generate an ASCII file of the test data. For each test case the data file covered a time span of one hour with five minute interval readings. Figure 6.1 shows an example record of this ASCII file. As mentioned above SAX requires the data in five second intervals. As well, process I/O variables require the generation of additional information as mentioned in Chapter 3. To perform these functions a two stage front end data processor was developed.

```
Time                                  YY-MM-DD/HH:MM:SS
Reheat Steam Temperature              F8.2
Main Steam Temperature                F8.2
Gas Exit Temperature                  F8.2
Steam Flow                            F8.2
Main Steam Pressure                   F8.2
Excess Air                            F8.2
Burner Tilt                           F8.2
Main Steam Attemperating Spray        F8.2
Reheat Steam Attemperating Spray      F8.2
```

Figure 6.1 LOTUS 123 - Record Format - SAX Raw Data Test File

In stage one of the data processor the five minute variable readings were linearly interpolated to produce a file with process I/O data every five seconds. Along with this, the time field in each record was converted from a time string to a numeric representation (described in Chapter 3). The process data was also normalized in this stage of the data processor (discussed in Chapter 3).

Stage two of the data preprocessor generated the additional information required by SAX for each process variable. This information included the error, error slope and average error slope of the variable.

6.4 ART-IM Interface to External Process I/O File

ART-IM has the capability to interface with external data files through the definition of an external data file record. This record defines the conversion of external data to ART-IM objects. For the SAX system an external data definition record was created to read the process I/O file. This record definition is shown in Figure 6.2.

The SAX system reads the process data file on a periodic basis. The rule which initiates the reading of the data file in SAX has a salience of -5 which means that it is fired only when all the rules on the agenda which have a higher salience have fired.

```
def-external-data pio_fact
      " process io record defined as a fields" 424
      ( time                 0   10  :char   :string )
      ( time_string          10  17  :char   :string )
      ( reheat_temp_val      26  7   :char   :float )
      ( norm_reheat_temp_val 35  9   :char   :float )
      ( reheat_temp_setpt    44  9   :char   :float )
      ( reheat_temp_error    53  9   :char   :float )
      ( reheat_temp_slope    62  9   :char   :float )
      ( main_temp_val        71  8   :char   :float )
      ( norm_main_temp_val   79  9   :char   :float )
      ( main_temp_setpt      88  9   :char   :float )
      ( main_temp_error      97  9   :char   :float )
      ( main_temp_slope      106 9   :char   :float )
      ( gas_temp_val         115 8   :char   :float )
      ( norm_gas_temp_val    123 9   :char   :float )
      ( gas_temp_setpt       132 9   :char   :float )
      ( gas_temp_error       141 9   :char   :float )
      ( gas_temp_slope       150 9   :char   :float )
      ( steam_flow           159 8   :char   :float )
      ( norm_steam_flow      167 9   :char   :float )
      ( steam_flow_setpt     176 9   :char   :float )
      ( steam_flow_error     185 9   :char   :float )
      ( steam_flow_slope     194 9   :char   :float )
      ( main_press           203 8   :char   :float )
      ( norm_main_press      211 9   :char   :float )
      ( main_press_setpt     220 9   :char   :float )
      ( main_press_error     229 9   :char   :float )
      ( main_press_slope     238 9   :char   :float )
      ( excess_air           247 8   :char   :float )
      ( norm_excess_air      255 9   :char   :float )
      ( excess_air_setpt     264 9   :char   :float )
      ( excess_air_error     273 9   :char   :float )
      ( excess_air_slope     282 9   :char   :float )
      ( brn_tilt             291 8   :char   :float )
      ( norm_brn_tilt        299 9   :char   :float )
      ( brn_tilt_setpt       308 9   :char   :float )
      ( brn_tilt_error       317 9   :char   :float )
      ( brn_tilt_slope       326 9   :char   :float )
      ( main_spray           335 8   :char   :float )
      ( norm_main_spray      343 9   :char   :float )
      ( main_spray_setpt             :char   :float )
      ( main_spray_error     361 9   :char   :float )
      ( main_spray_slope     370 9   :char   :float )
      ( reheat_spray         379 8   :char   :float )
      ( norm_reheat_spray    387 9   :char   :float )
      ( reheat_spray_setpt   396 9   :char   :float )
      ( reheat_spray_error   405 9   :char   :float )
      ( reheat_spray_slope   414 9   :char   :float )
```

Figure 6-2 SAX Process I/O External Data File Record

The majority of rules in SAX are at the default salience which is 0. This indicates that the reading of process I/O only occurs if there is nothing else on the agenda.

Once the process I/O data file is read the temporal preprocessor begins processing the data (described in Chapter 3).

6.5 SAX Tests

There were five test data files created to exercise the SAX sootblowing advice capabilities. These files are found in Appendix V. As discussed, these tests were chosen based on sootblowing actions in the plant during the two week period that sootblowing data was recorded. They represented together, the majority of reasons operators cited for performing sootblowing operations.

As a result of running SAX , sootblowing strategies were chosen. These strategies specified individual sootblowers which were to be operated together to reduce the slag or dust buildup in the boiler. By so doing, improved heat absorption would be achieved.

Table 6-1 summarizes the tests and results. In this table the actual operator sootblowing actions and reasons are presented. Along with this, the sootblowing actions and associated reasons chosen by SAX are given. For the purpose of interpreting the sootblowing reasons cited by the operators the following list is given.

1 -	Gas Exit Temperature High
2 -	Temperatures High
3 -	Air Heater Maintenance
4 -	Boiler Observations
5 -	Balance Temperatures
6 -	Economizer Maintenance
7 -	Primary Maintenance
8 -	Reheat Steam Temperature High
9 -	Main Steam Temperature High
10 -	Eliminate Main Spray
11 -	Eliminate Reheat Spray
12 -	Burner Tilt Low
13 -	Temperatures Low

A complete discussion of the test results are included in Appendix V with the associated test data file and SAX screens. Following is a brief analysis of the tests performed.

For SAX test #1, the operator cited the main steam, reheat steam and gas exit temperatures high as reasons for operating the specified sootblowers. SAX detected low main and reheat steam temperatures and a high gas exit temperature from the test data resulting in selecting the appropriate strategies to improve the heat absorptions.

For SAX test #2 the operator felt the reheat steam temperature was high. SAX detected the main steam temperature and gas exit temperatures high with excess air moving from normal to high. As a result, SAX chose waterwall sootblowing strategies to reduce the radiant heat absorption.

For SAX test #3 the operator detected a high gas exit temperature combined with low main and reheat steam temperatures. SAX detected similar conditions and chose strategies to improve the heat absorption in the radiant areas of the boiler.

For the fourth SAX test the operator cited gas exit and main steam temperatures high with the requirement to eliminate the use of main steam attemporating spray. SAX also detected this condition along with other plant states which required sootblowing actions. It should be noted here that the plant had just been brought online after a boiler trip had occurred (unexpected plant shutdown). This probably explains the large number of sootblowing operations recommended by the operator. SAX also noted a significant number of conditions in the test data on which to recommend sootblowing operations.

TEST #	SOOTBLW ACTION/ REASONS	SOOTBLW DATE/ TIME SPAN	OPERATOR STRATEGIES	SAX RULES/ SOOTBLOWING STRATEGIES
1	Action 12 1,3,5,7	01/26/90 21:30-22:30	Prim Sshtr Uww Mww Ahtr	PS_1 - Mww PS_18 - Prim Sshtr Maint - Prim Econo Ahtr
2	Action 22 3,8	01/28/90 21:30-22:30	Uww Mww Ahtr	PS_8 - Prim PS_4 - Mww Lww Maint - Prim Econo Ahtr
3	Action 27 1,13	01/29/90 12:30-13:30	Prim Frhtr Hrhtr	PS_18 - Prim Sshtr Mww
4	Action 46 1,3,9,10	02/02/90 8:30-9:30	Prim Frhtr Hrhtr Uww Mww	PS_1 - Frhtr Mww PS_9 - Prim Sshtr PS_17 - Lww PS_5 - Prim Mww
5	Action 49 3,4,7	02/03/90 2:30-3:30	Prim Slope Ahtr	PS_1 - Frhtr Mww PS_20 - Prim Mww

Table 6-1 SAX Test Summary

For SAX test #5, the operator cited regular maintenance and observations as the reasons for the chosen sootblowing strategies. SAX, however detected high reheat steam, main steam and gas exit temperatures at one point in the test (plant state 20) and low reheat and main steam temperatures at another point in the test (plant state 1). Appropriate sootblowing strategies were chosen to improve the heat absorptions where required.

The analysis shows that the operator sootblowing actions chosen do not directly correspond to the SAX sootblowing actions. This is related to the inaccuracies of the operator data provided from the plant as well as difficulties in the collection of the data from the strip charts and logs. Given the existing knowledge base, SAX makes correct sootblowing decisions for the situations arising in the test data. The principal reason for different sootblowing decisions by the operator is the difference in the data seen by SAX compared to what the operator actually saw. For example, the reheat steam temperature in test 1 was recorded as low whereas the strip charts had this temperature recorded as high. This is also noted in section 6.2 above, and is supported by the data recorded in Appendix IV.

A further validation of SAX's knowledge base was done by showing the rules to a senior operator with 25 years experience in various power plants. He made several recommendations for rule improvements, but was in agreement with most of the SAX rules.

7 CONCLUSIONS

7.1 Overview of the Thesis Work

This thesis deals primarily with the extension of expert systems to consistently handle real-time information typical of process control systems. A survey of current literature on *real-time systems* and *real-time expert systems*, in particular, provided an understanding of the issues involved. This lead to an in-depth study of approaches to temporal reasoning which formed the theoretical basis of this thesis. As a result, the event calculus model of temporal reasoning originally proposed by Kowalski was chosen for implementation.

A complete domain independent temporal processing system was developed for use in a forward chaining expert system shell. This includes (1) a temporal processor to identify significant events and initiate and terminate temporal relationships based on these events, (2) a temporal history processor to store and retrieve past temporal information and (3) a query processor to transparently handle requests from the expert system for either current or historical temporal relationship information.

The temporal processor handles analog temporal data typical of a process control environment as well as operator generated data received through an interactive CRT based user interface. The analog data included three generic types called setpoint variables, process variables and control device variables. The operator generated data included the entry of such information as boiler observation and boiler configuration changes.

A prototype sootblowing advisory expert (SAX) was built using this methodology. Specific rules about sootblowing at the Dalhousie Unit #2 coal-fired generating station were captured and are specified in Appendix III. This system comprises 140 rules with approximately 12,600 lines of ART-IM code. It runs on a IBM PS/2 model 70.

Five sample one hour tests with interpolated plant data were performed to validate SAX's operation. These tests showed that SAX will correctly respond to the a subset of the sootblowing decisions required at the Dalhousie plant. Further testing at the plant would complete the validation of the SAX system.

7.2 **Future Work**

The future work can be broadly classified in two areas; (1) Temporal Logic Development and (2) Expansions of the SAX prototype to a production quality expert system.

For this thesis event calculus was developed based on the assumption that all events occur chronologically. This could be further developed to handle events which occur in any order. This encompasses the need to efficiently handle the ramification, qualification and frame problems which occur in time-based approaches of temporal reasoning.

The SAX prototype can be expanded and improved in several areas. The knowledge base can be improved by adding new rules to handle different plant loads, different boiler types, and different fuel types. The user interface could be further developed to achieve the quality necessary for plant operation. This would involve changing the interface from its current textual format to a graphical format. The query processor could be expanded to provide more query types. An example would be a query of a temporal object over a range of time
(query object time1 time2). The SAX system could also be deployed at the Dalhousie plant by connecting to actual process I/O which would provide a realistic

test of the proper operation of SAX by the intended users of the system.

In my opinion the SAX prototype is worth further development. Plans are being made to incorporate SAX into the Improved Steam Temperature Control System currently being developed at Maritime Nuclear.

REFERENCES

ALLE84	Allen, James F., "Towards a General Theory of Action and Time", Artificial Intelligence Vol.23 1984, pp.123-154.
BALA89	Balasubramanyan, Jaishree, "A Historical Relational Database Management System", Masters Thesis, School of Computer Science, University of New Brunswick, N.B., May 1989.
BORL89a	Borland International, Turbo Prolog 2.0 Reference Guide. Borland International, Scotts Valley, CA 95066-0001, 1988.
BORL89b	Borland International, Turbo Prolog 2.0 User's Guide. Borland International, Scotts Valley, CA 95066-0001, 1988.
DUKE86	Dukelow, Sam G., The Control of Boilers. Instrument Society of America, 1986.
KOWA85	Kowalski, Robert and Sergot, Marek, "A Logic-based Calculus of Events", New Generation Computing Vol.4, 1986, pp.67-95.
KOWA86	Kowalski, Robert, Database Updates in the Event Calculus, Doc 86/21, Imperial College London, July 1986.
LAFF88	Laffey, Thomas J., Cox, Preston A., Schmidt, James L. et. al., "Real-Time Knowledge Based Systems". AI Magazine, Vol.9, No. 1, Spring 1988, pp.27-45.
MARI90	Maritime Nuclear, "An Advanced Steam Temperature Controller In Coal-Fired Generating Units", 522 G 623, Canadian Electrical Association, February 1990.
MCDE80	McDermott, Drew and Doyle, Jon, "Non-Monotonic Logic I", Artificial Intelligence Vol.13 1980, pp.41-72.
MCDE82	McDermott, Drew, "A Temporal Logic for Reasoning about Processes and Plans", Cognitive Science Vol.6, 1982, pp.101-155.
ROLS88	Rolston, David W., Principles of Artificial Intelligence and Expert Systems Development. McGraw-Hill Book Company, 1988.

SHOH88 Shoham, Yoay and Goyal, Nita, "Temporal Reasoning in Artificial Intelligence", In Exploring AI: Survey Talks from the National Conferences, Howard E. Strobe and Morgan Kaufman, 1988, pp.419-438.

STAN88 Stankovic, John A., "Real-Time Computing Systems: The Next Generation". Coins Technical Report, Department of Computer and Information Science, University of Massachusetts, Amherst, Ma., June 1988.

STEP89 Stephens, K. and Nickerson, B., "Temporal Reasoning Considerations for a Sootblowing Advisor", Proceedings of the 2nd UNB Applications of AI Workshop, Fredericton, N.B., Oct 3, 1989, pp.18-28.

TRUD89a Trudel, André, "The Interval Representation Problem", To appear in: International Journal of Intelligent Systems, special volume on temporal issues in AI, August 1991.

TRUD89b Trudel, André and Goodwin, Scott D., Persistence in Continuous First Order Temporal Logics, TR89-24, Department of Computer Science, University of Alberta, Edmonton, Alberta, Canada, October 1989.

TRUD89c Trudel André, "A First Order Logic For Qualitative and Quantitative Domains", APICS Annual Computer Science Conference Proceedings, Fredericton, N.B., 5 November 1989, pp.15-23.

TRUD90 Trudel, André, "Temporal Integration", in: Proceedings of Eight Biennial Conference of the Canadian Society for Computational Studies of Intelligence, May 1990, Ottawa, Canada, pp40-45.

WYNN79 Wynnyckyj, J.R., Chambers, A.K., Rhodes, E. "Fouling Dynamics In Furnaces of Large Pulverized Coal Fired Boilers". Department of Chemical Engineering, University of Waterloo, 1980.

WYNN80 Chambers, A.K., Wynnyckyj, J.R., Rhodes, E. "A Furnace Wall Ash Monitoring System for Coal Fired Boilers", The American Society of Mechanical Engineers, November 1980.

BIBLIOGRAPHY

BLAC 86	Black, W.J., Intelligent Knowledge Based Systems, An Introduction. Van Nostrad Reinhold (U.K.) Co. Ltd., Berkshire, England, 1986.
FLOD88	Floyd, Michael, "Suitable For Framing". Turbo Technix. March/April 1988, pp.80-87.
HOFF87	Hoffman, Robert R., "The Problem of Extracting the Knowledge of Experts from the Perspective of Experimental Psychology". AI Magazine., Vol8. No2, Summer 1987, pp.53-67.

APPENDIX I

PLANT OPERATIONS INTERVIEW

APPENDIX I

PLANT OPERATIONS INTERVIEW

INTERVIEW QUESTIONS

1. Is there a difference in operation when the burner tilt changes? Is the following true? When the gain term is over 30° positive tilt there will be a rapid change in steam temperature to a small change in burner tilt. When the burner tilt is negative the change in steam temperature will be flat.
2. What specific conditions in the plant cause you to take action, and what action?
3. Under what circumstances do you anticipate problems in maintaining tight control on steam temperature (ie. when is steam temperature control easy and when is it hard)?
4. How good is the soot monitoring (ie. heat absorption method) at the Dalhousie Plant? Does it represent the conditions in the boiler? How does the heat absorption method indicate slagging conditions in the boiler?
5. How fast do circumstances change in the boiler with regards to slag buildup?
6. What criteria is used to operate the sootblowers?
7. Is there other information that contributes to your decision for controlling steam temperature (ie. high positive burner tilt, heat absorption in the superheater and reheater receives little change, then the superheater and reheater region is more highly fouled)?
8. There are small detail differences in a boiler design. What approach would you say is specific to the Dalhousie boiler in controlling the reheat temperatures?
9. Where are sootblowers? How are they controlled? How are they operated?

10. What information does the operator have available on which to base his decisions for steam temperature control as well as sootblowing?

11. How does the operator know once he has taken an action (ie. operated sootblowers) that his control of the situation is working?

12. Under what circumstances is the decision to switch from automatic to manual control and vice versa made?

13. Do you allow fouling to occur in certain areas? If so, where? Why?

14. Do you receive information from temperatures in the flue gas?

15. What fouling characteristics have been observed with the different coals being burned?

16. Are the operators aware when the boiler sheds large amounts of slag rapidly? What seems to cause this cleaning phenomena? Do you try to avoid this boiler cleaning phenomena or do you just ride out the large transient in steam temperature that occurs? What control actions are taken, if any?

17. How do you control while you are sootblowing? What is the effect of the control while sootblowing? Is the control switched from auto to manual?

INTERVIEW ANSWERS

1. Dalhousie Unit #2 is a Combustion Engineering boiler with Bailey Controls. At full power the reheat steam temperature can be basically controlled by burner tilt. Different slagging conditions would result in different combinations of burner tilt and therefore, there is more than one gain curve for burner tilt. There is a limiting of burner tilts at minimum load. Below 50% load, the tilt is limited to 65% of the signal. (50% = 0° tilt). This occurs because the flame scanners will detect a loss of flame and trip as the fire ball will have been moved above the top of their range. If the burner tilt is down the fire ball is not as clean, a lot of smoking occurs.

2. The following conditions in the plant require action:
   - Steam Temperature is too low penalized in dollars
   - Spraying too much attemperator water counteract by blowing soot.
   - Burner tilt is out of range +80 and above or -30 and below.
   - Negative burner tilt indicates to much heat in the superheater region or the waterwalls are dirty.

3. The tight control of steam temperature is hindered when:
   - Change over of mills occurs
   - Sootblowing occurs
   - High degree of slagging (ash content 18%-29%)
     which occurs when the ash content is high which causes the heat content of the fuel to decrease. Moisture and heat content (BNI) are measured once a week.

4. The Computer Plant Technology System (CATS) performs boiler efficiency calculations as well as heat absorption calculations. The CATS system uses heat absorptions as an indication of slag buildup. The calculated heat absorptions are not necessarily indicative of slagging conditions since the target heat absorptions are not set correctly. In CATS, the mill arrangements, as well as burner tilts, are not included in the calculations. The heat absorption method can indicate problems in the waterwalls, superheater, primary and economizer regions. A visual inspection can confirm this and thus indicate the need for a sootblowing operation. However, the calculated heat absorptions are not necessarily indicative of slagging since the targets which are used for comparisons to detect irregular conditions have not been determined correctly.

5. Transients in temperature occur depending on slagging conditions. Steam temperature moves quickly when certain changes occur.It is possible to see a 20° change in temperature if a wall sootblowing action effectively moved a lot of slag.A good operator would adjust burner tilt first before the wall blowing to accommodate for the transients which will occur.

6. Slagging occurs depending on the combustion of fuel when combined with different burner tilts, spray, coal and mill arrangements. Sets of blowers are common to both the reheat and superheat regions which requires a judgement on whether to sootblow. You may not want to affect the temperature in one region. Sootblowing occurs more in the primary superheater region because of the higher convection action. When reheat spray is used then sootblowing would be required in the primary since the heat was too high in the reheat region.

7. A percent change in excess air has a greater effect than a percent change in load. An increase in excess air increases the gas exit temperature. Steam temperature control is governed by tilt, excess air, load, wall cleaning and coal quality. It is more difficult to control reheat temperature thus operators blow soot to attempt to reduce the temperature.

8. At full load, when burner tilts are high, to maintain reheat temperature the use of spray will be required. Except on the coldest months you might use less excess air to reduce the need for reheat sprays. It is necessary to watch the outside air temperature. If there is not enough heat in the flue gasses you will lose 250° at the low end. When attemporating spray is used in the superheater region a response in temperatures occurs in about 3-4 minutes/7-8 minutes. Reheat spray has a longer response time. Sootblowing in the waterwall region takes 3 minutes with a 3 minute respond before the steady state is obtained again. The secondary section is a radiant heat region in the boiler. The primary superheat is both radiant and convection. This modifies with load. When the reheat and superheater temperatures are down adjust burner tilts, adjust excess air and bring sprays off auto (generally on auto).

9. It is possible to modulate sequences of sootblowers or operate them all separately. Sootblowers blow steam when they are retracting into the wall, not when they are on their way into the boiler. Superheaters are shielded to cut down on erosion during sootblowing. Repetitive sootblowing in the same areas of the boiler will cause erosion and is considered to be a big problem. The sootblowers are checked every shift. The burners are checked every twenty-four hours for possible

clogging. An operator requires two indicators of slag buildup before he will perform a sootblowing operation.

10. Sootblowing is based on a visual inspection of the boiler interior. As well as visual inspections, sootblowing needs are determined from gas exit temperatures and burner tilts. Under 4 mill operation there is more slagging in the bottom regions of the boiler. Slag also builds up in the cavities between the superheater banks.

11. Once an operator has instituted a control or sootblower action he must watch the available plant information and access from experience whether the action was successful. It is possible that the action taken does not produce the results required in a timely fashion.

12. When the controls are on manual and the load is increased, the steam temperature increases because with more fuel there is more gas flow. Burner tilts are always on manual and when the mill or air configuration changes.

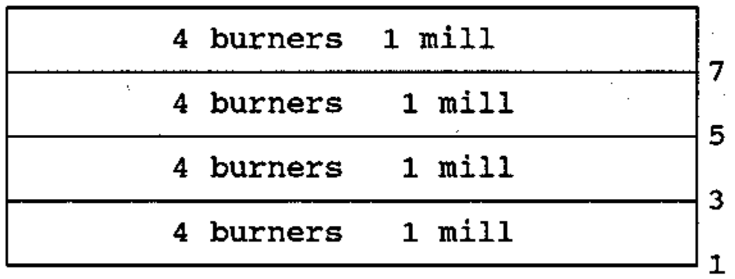
13. Dalhousie is dependent on a certain amount of slag in the boiler. The boiler is seasoned on startup after the annual Plant Shutdown by not blowing soot for a day to a day and a half.

14. Gas Exit temperature should be maintained at 660° on the furnace gas out before the air heaters (boiler exit out). If the gas exit temperature reaches 660° then this indicates that fouling has occurred and sootblowing is needed.

15. Minto Coal, which is normally burned at Dalhousie, is a high slagging coal. Dalhousie Unit #2 was designed to burn Minto coal. In the winter months four mills are operated and air flow changes. During

normal operation three mills are used. Coal characteristics such as moisture content or mud content generally requires that more coal to be used (ie. a need for more mills). There is a problem with weekend coal since it comes from the stock pile rather than fresh and thus tends to block mills and burners and increases slagging problems (indicates the need to modify rules). Mills are overhauled every two months. Minto coal contains 20-23% ash. Nova Scotia coal has 10% ash, it's cleaner and has less mill wear, less soot but has too much heat in reheat region. Columbia coal causes the boiler to self clean over a period of two days. There seems to be little self cleaning with Minto coal.

Configuration of Burners



16. Yes the operators are aware of sheets of slag dropping from the boiler walls. They attempt to deal with the large transient in steam temperature that occurs using standard control procedures. If the slag buildup is thick reducing the load (any major disturbance) will cause the slag to fall away (self cleaning effect). Reducing the load helps to remove the slag buildup at the bottom of the boiler (ie. the slopes). Self cleaning will occur if a large pressure, temperature, load changes,

or upset condition occurs. Sootblowing occurs in the superheater banks to obtain more heat absorption. Changing the mill setpoint to change the BTU input into furnace (2 mins longer). A transport delay occurs when you change the output to the pulverizer.

17. This is dependent on the current situation and the operator involved.

APPENDIX II

SOOTBLOWING DATA ACQUISITION PROCEDURE

APPENDIX II

SOOTBLOWING DATA ACQUISITION PROCEDURE

PROCEDURES FOR THE SOOTBLOWING SUMMARY SHEET

When an operator is about to perform a sootblowing operation could he/she please fill out a **SOOTBLOWING SUMMARY SHEET.** On this sheet you will find the following items:

1. **BEFORE** - The variables listed under the column BEFORE are to be filled out just before the sootblowing action is started. The Time entry is the time the sootblowing action was started. On the Mills In Operation entry please give the list of all mills currently operating. All other entries are as stated (ie. Reheat Steam Temp. is the Reheat Steam Temperature before the sootblowing action is started).
2. **AFTER** - The same variables are to be recorded after the sootblowing action has completed and the control system has settled. If you initiate another sootblowing action before the system has settled then record the readings before you start the next sootblowing action. Record the time these variables readings are taken.
3. **SOOTBLOWERS OPERATED** - In this section please list the sootblowers activated and why you chose these sootblowers.
4. **SOOTBLOWING REASONS** - In this section please state the reasons you have decided to sootblow and the results you expect to see from the sootblow.

SOOTBLOWING SUMMARY SHEET

VARIABLES	BEFORE	AFTER
Time		
Main Steam Temp.		
Reheat Steam Temp.		
Main Steam Press.		
Main Steam Spray		
Reheat Steam Spray		
Burner Tilt		
Excess Air		
Furnace Gas Exit Temp.		
Mills In Operation		

SOOTBLOWERS OPERATED

SOOTBLOWING REASONS

APPENDIX III

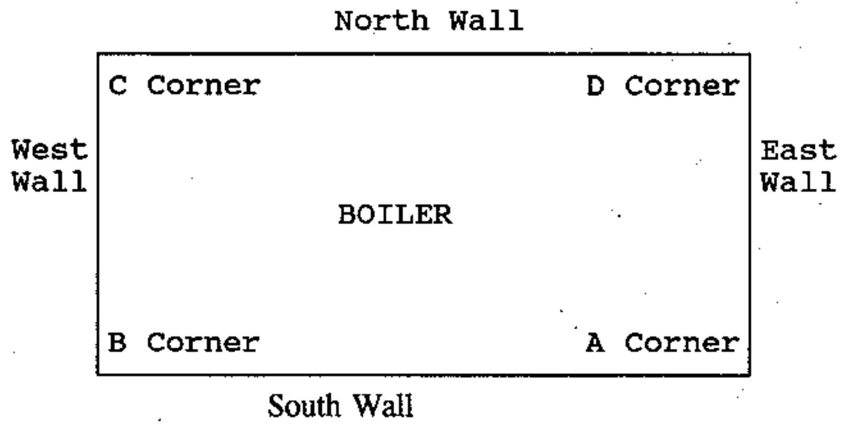
SOOTBLOWING ADVISOR RULES

APPENDIX III

SOOTBLOWING ADVISOR RULES

These rules were prepared for implementation in the SAX system. The first section provides background information on boiler orientation and sootblower distribution in the boiler.

BOILER ORIENTATION



SOOTBLOWER DISTRIBUTION

BOILER SECTION	SOOTBLOWERS
Slopes	
North	IR67, IR63, IR66
East	IR65, IR61, IR68
Lower Waterwalls	
North	IR7, IR8, IR9
East	IR4, IR5, IR6
South	IR1, IR2, IR3
West	IR10, IR11, IR12

Middle Waterwalls

North	IR19, IR20, IR21, IR31, IR32, IR33
East	IR16, IR17, IR18, IR28, IR29, IR30
South	IR13, IR14, IR15, IR25, IR26, IR27
West	IR22, IR23, IR24, IR34, IR35, IR36

Upper Waterwalls

North	IR43, IR44, IR45
East	IR40, IR41, IR42
South	IR37, IR38, IR39
West	IR46, IR47, IR48

Secondary Superheater

East	IK1, IK3, IK5, IK7
West	IK2, IK4, IK6, IK8

Cold Reheat

East	IK9, IK11, IK13, IK15
West	IK10, IK12, IK14, IK16

Hot Reheat

East	IK17, IK19, IK21, IK23
West	IK18, IK20, IK22, IK24

Final Reheat

East	IK25, IK27, IK29, IK31
West	IK26, IK28, IK30, IK32

Primary

East	IK35, IK37, IK39, IK41
West	IK36, IK38, IK40, IK42

Economizer	IK33, IK34
Air Heater	IKAH1, IKAH2, IKAH3, IKAH4

RULE TYPE: Select Sootblowers

Select sootblowing strategies for the areas in which slag has been detected.

```
1.    IF
            slag on slopes AND
      THEN
            from the list of available slope_sootblowers choose the number of
            slope_sootblowers depending on the slag type AND
            display the sootblowing advice from the sootblow_list with appropriate
            reasoning.

2.    IF
            slag on lower_waterwalls
      THEN
            from the list of available lower_waterwalls_sootblowers choose the
            number of lower_waterwall_sootblowers depending on the slag type
            AND
            display the sootblowing advice from the sootblow_list with appropriate
            reasoning.

3.    IF
            slag on middle_waterwalls
      THEN
            from the list of available middle_waterwalls_sootblowers choose the
            number of middle_waterwall_sootblowers depending on the slag type
            AND
            display the sootblowing advice from the sootblow_list with appropriate
            reasoning.

4.    IF
            slag on upper_waterwalls
      THEN
            from the list of available upper_waterwalls_sootblowers choose the
            number of sootblowers depending on the slag type AND
            display the sootblowing advice from the sootblow_list with appropriate
            reasoning.
```

5. IF

   slag on secondary_superheater

   THEN

   from the list of available secondary_superheater_sootblowers choose the number of sootblowers depending on the slag type AND
   display the sootblowing advice from the sootblow_list with appropriate reasoning.

6. IF

   slag on cold_reheat

   THEN

   from the list of available cold_reheat sootblowers choose the number of cold_reheat_sootblowers depending on the slag type AND
   display the sootblowing advice from the sootblow_list with appropriate reasoning.

7. IF

   slag on hot_reheat

   THEN

   from the list of available hot_reheat sootblowers choose the number of hot_reheat_sootblowers depending on the slag type AND
   display the sootblowing advice from the sootblow_list with appropriate reasoning.

8. IF

   slag on final_reheat

   THEN

   from the list of available final_reheat_sootblowers choose the number of final_reheat_sootblowers depending on slag type AND
   display the sootblowing advice from the sootblow_list with appropriate reasoning.

9. IF

   slag on primary

   THEN

   from the list of available primary_sootblowers choose the number of primary_sootblowers depending on slag_type AND
   display the sootblowing advice from the sootblow_list with appropriate reasoning.

RULE TYPE : Slag On Manual Observation

Slag accumulation is described by two parameters, the type and amount of buildup. The type of buildup is either dust or slag. The amount is described by the inches of accumulation.

```
10.   IF
                slag is manually observed on boiler_slopes
      THEN
                assert observe_slag_boiler_slopes.

11.   IF
                slag is manually observed on boiler_lower_waterwalls
      THEN
                assert observe_slag_lower_waterwalls.

12.   IF
                slag is manually observed on boiler_middle_waterwalls
      THEN
                assert observe_slag_middle_waterwalls.

13.   IF
                slag is manually observed on boiler_upper_waterwalls
      THEN
                assert observe_slag_middle_waterwalls.

14.   IF
                slag is manually observed on boiler_secondary_superheater
      THEN
                assert observe_slag_secondary_superheater.

15.   IF
                slag is manually observed on boiler_cold_reheat
      THEN
                assert observe_slag_cold_reheat.

16.   IF
                slag is manually observed on boiler_hot_reheat
      THEN
                assert observe_slag_hot_reheat.
```

17. IF
    slag is manually observed boiler_final_reheat
THEN
    assert observe_slag_final_reheat.

18. IF
    slag is manually observed boiler_primary
THEN
    assert observe_slag_primary.

RULE TYPE : Hold On Selection Of A Sootblowing Strategy

19. IF
    a sootblowing action has been initiated
THEN
    wait an appropriate amount of time for the control system to reach steady state before determining if further sootblowing actions are required.

RULE TYPE : Determination Of Slag

20. **PLANT_STATE_1**

```
IF the plant is operating at full load AND
   the control_system_stable_timer is zero AND
   this plant state has been in existence for a minimum of time
   determined by the PLANT_STATE_1 timer AND
   Reheat_Steam_Temp is in the low range AND
   Main_Steam_Temp is in the low range AND
   Gas_Exit_Temp is in the high range AND
   Burner_Tilt is in the low range AND
   Excess_Air is in the normal range AND
   Reheat_Steam_Spray is in the normal range AND
   Main_Steam_Spray is in the normal range
THEN
   assert slag_final_reheat_section AND
   assert slag_middle_waterwall_section
```

21. **PLANT_STATE_2**

```
IF the plant is operating at full load AND
   the control_system_stable_timer is zero AND
   this plant state has been in existence for a minimum of time
   determined by the PLANT_STATE_2 timer AND
   Reheat_Steam_Temp is in the norm range AND
   Main_Steam_Temp is in the norm range AND
   Gas_Exit_Temp is in the high range AND
   Burner_Tilt is in the low range AND
   Excess_Air is in the normal range AND
   Reheat_Steam_Spray is in the normal range AND
   Main_Steam_Spray is in the normal range
THEN
   assert slag_primary_section AND
   assert slag_secondary_section AND
   assert slag_upper_waterwall_section
```

22. **PLANT_STATE_3**

```
IF the plant is operating at full load AND
   the control_system_stable_timer is zero AND
   this plant state has been in existence for a minimum of time
   determined by the PLANT_STATE_3 timer AND
   Reheat_Steam_Temp is in the high range AND
   Main_Steam_Temp is in the norm range AND
   Gas_Exit_Temp is in the high range AND
   Burner_Tilt is in the low range AND
   Excess_Air is in the normal range AND
   Reheat_Steam_Spray is in the normal range AND
   Main_Steam_Spray is in the normal range
THEN
   assert slag_primary_section AND
   assert slag_upper_waterwall_section AND
   assert slag_middle_waterwall_section
```

23. **PLANT_STATE_4**

```
IF the plant is operating at full load AND
   the control_system_stable_timer is zero AND
   this plant state has been in existence for a minimum of time
   determined by the PLANT_STATE_4 timer AND
   Reheat_Steam_Temp is in the normal range AND
   Main_Steam_Temp is in the high range AND
   Gas_Exit_Temp is in the high range AND
   Burner_Tilt is in the low range AND
   Excess_Air is in the normal range AND
   Reheat_Steam_Spray is in the normal range AND
   Main_Steam_Spray is in the normal range
THEN
   assert slag_middle_waterwall_section AND
   assert slag_lower_waterwall_section
```

24. **PLANT_STATE_5**

```
IF the plant is operating at full load AND
   the control_system_stable_timer is zero AND
   this plant state has been in existence for a minimum of time
   determined by the PLANT_STATE_5 timer AND
   Reheat_Steam_Temp is in the normal range AND
   Main_Steam_Temp is in the normal range AND
   Gas_Exit_Temp is in the high range AND
   Burner_Tilt is in the low range AND
   Excess_Air is in the normal range AND
   Reheat_Steam_Spray is in the normal range AND
   Main_Steam_Spray is in the high range
THEN
   assert slag_primary_section
   assert slag_middle_waterwall_section
```

25. **PLANT_STATE_6**

```
IF the plant is operating at full load AND
   the control_system_stable_timer is zero AND
   this plant state has been in existence for a minimum of time
   determined by the PLANT_STATE_6 timer AND
   Reheat_Steam_Temp is in the normal range AND
   Main_Steam_Temp is in the normal range AND
   Gas_Exit_Temp is in the high range AND
   Burner_Tilt is in the low range AND
   Excess_Air is in the high range AND
   Reheat_Steam_Spray is in the normal range AND
   Main_Steam_Spray is in the high range
THEN
   assert slag_middle_waterwall_section
```

26. **PLANT_STATE_7**

```
IF the plant is operating at full load AND
   the control_system_stable_timer is zero AND
   this plant state has been in existence for a minimum of time
   determined by the PLANT_STATE_7 timer AND
   Reheat_Steam_Temp is in the high range AND
   Main_Steam_Temp is in the high range AND
   Gas_Exit_Temp is in the normal range AND
   Burner_Tilt is in the low range AND
   Excess_Air is in the high range AND
   Reheat_Steam_Spray is in the normal range AND
   Main_Steam_Spray is in the normal range
THEN
   assert slag_cold_reheat_section AND
   assert slag_secondary_section AND
   assert slag_upper_waterwall_section
```

27. **PLANT_STATE_8**

```
IF the plant is operating at full load AND
   the control_system_stable_timer is zero AND
   this plant state has been in existence for a minimum of time
   determined by the PLANT_STATE_8 timer AND
   Reheat_Steam_Temp is in the normal range AND
   Main_Steam_Temp is in the high range AND
   Gas_Exit_Temp is in the high range AND
   Burner_Tilt is in the low range AND
   Excess_Air is in the high range AND
   Reheat_Steam_Spray is in the normal range AND
   Main_Steam_Spray is in the normal range
THEN
   assert slag_primary_section AND
   assert slag_lower_waterwall_section
```

28. **PLANT_STATE_9**

```
IF the plant is operating at full load AND
   the control_system_stable_timer is zero AND
   this plant state has been in existence for a minimum of time
   determined by the PLANT_STATE_9 timer AND
   Reheat_Steam_Temp is in the low range AND
   Main_Steam_Temp is in the normal range AND
   Gas_Exit_Temp is in the high range AND
   Burner_Tilt is in the low range AND
   Excess_Air is in the normal range AND
   Reheat_Steam_Spray is in the normal range AND
   Main_Steam_Spray is in the normal range
THEN
   assert slag_hot_reheat_section
   assert slag_secondary_section
```

29. **PLANT_STATE_10**

```
IF the plant is operating at full load AND
   the control_system_stable_timer is zero AND
   this plant state has been in existence for a minimum of time
   determined by the PLANT_STATE_10 timer AND
   Reheat_Steam_Temp is in the low range AND
   Main_Steam_Temp is in the normal range AND
   Gas_Exit_Temp is in the high range AND
   Burner_Tilt is in the high range AND
   Excess_Air is in the normal range AND
   Reheat_Steam_Spray is in the normal range AND
   Main_Steam_Spray is in the normal range
THEN
   assert slag_final_reheat_section
   assert slag_cold_reheat_section
```

30. **PLANT_STATE_11**

```
IF the plant is operating at full load AND
   the control_system_stable_timer is zero AND
   this plant state has been in existence for a minimum of time
   determined by the PLANT_STATE_11 timer AND
   Reheat_Steam_Temp is in the normal range AND
   Main_Steam_Temp is in the normal range AND
   Gas_Exit_Temp is in the high range AND
   Burner_Tilt is in the normal range AND
   Excess_Air is in the normal range AND
   Reheat_Steam_Spray is in the normal range AND
   Main_Steam_Spray is in the normal range
THEN
   assert slag_lower_waterwall_section
```

31. **PLANT_STATE_12**

```
IF the plant is operating at full load AND
   the control_system_stable_timer is zero AND
   this plant state has been in existence for a minimum of time
   determined by the PLANT_STATE_12 timer AND
   Reheat_Steam_Temp is in the low range AND
   Main_Steam_Temp is in the low range AND
   Gas_Exit_Temp is in the normal range AND
   Burner_Tilt is in the high range AND
   Excess_Air is in the high range AND
   Reheat_Steam_Spray is in the normal range AND
   Main_Steam_Spray is in the normal range
THEN
   assert slag_primary_section AND
   assert slag_final_reheat_section AND
   assert slag_middle_waterwall_section
```

32. **PLANT_STATE_13**

```
IF the plant is operating at full load AND
   the control_system_stable_timer is zero AND
   this plant state has been in existence for a minimum of time
   determined by the PLANT_STATE_13 timer AND
   Reheat_Steam_Temp is in the high range AND
   Main_Steam_Temp is in the high range AND
   Gas_Exit_Temp is in the high range AND
   Burner_Tilt is in the low range AND
   Excess_Air is in the low range AND
   Reheat_Steam_Spray is in the high range AND
   Main_Steam_Spray is in the high range
THEN
   assert slag_upper_waterwall_section AND
   assert slag_middle_waterwall_section AND
   assert slag_lower_waterwall_section
```

33. **PLANT_STATE_14**

```
IF the plant is operating at full load AND
   the control_system_stable_timer is zero AND
   this plant state has been in existence for a minimum of time
   determined by the PLANT_STATE_14 timer AND
   Reheat_Steam_Temp is in the high range AND
   Main_Steam_Temp is in the normal range AND
   Gas_Exit_Temp is in the high range AND
   Burner_Tilt is in the low range AND
   Excess_Air is in the normal range AND
   Reheat_Steam_Spray is in the high range AND
   Main_Steam_Spray is in the normal range
THEN
   assert slag_upper_waterwall_section AND
   assert slag_middle_waterwall_section
```

34. **PLANT_STATE_15**

```
IF the plant is operating at full load AND
   the control_system_stable_timer is zero AND
   this plant state has been in existence for a minimum of time
   determined by the PLANT_STATE_15 timer AND
   Reheat_Steam_Temp is in the normal range AND
   Main_Steam_Temp is in the normal range AND
   Gas_Exit_Temp is in the high range AND
   Burner_Tilt is in the low range AND
   Excess_Air is in the normal range AND
   Reheat_Steam_Spray is in the high range AND
   Main_Steam_Spray is in the normal range
THEN
   assert slag_upper_waterwall_section
```

35. **PLANT_STATE_16**

```
IF the plant is operating at full load AND
   the control_system_stable_timer is zero AND
   this plant state has been in existence for a minimum of time
   determined by the PLANT_STATE_16 timer AND
   Reheat_Steam_Temp is in the normal range AND
   Main_Steam_Temp is in the normal range AND
   Gas_Exit_Temp is in the high range AND
   Burner_Tilt is in the low range AND
   Excess_Air is in the normal range AND
   Reheat_Steam_Spray is in the high range AND
   Main_Steam_Spray is in the high range
THEN
   assert slag_primary_section AND
   assert slag_upper_waterwall_section
```

36. **PLANT_STATE_17**

```
IF the plant is operating at full load AND
   the control_system_stable_timer is zero AND
   this plant state has been in existence for a minimum of time
   determined by the PLANT_STATE_17 timer AND
   Reheat_Steam_Temp is in the normal range AND
   Main_Steam_Temp is in the high range AND
   Gas_Exit_Temp is in the high range AND
   Burner_Tilt is in the low range AND
   Excess_Air is in the normal range AND
   Reheat_Steam_Spray is in the normal range AND
   Main_Steam_Spray is in the high range
THEN
   assert slag_middle_waterwall_section AND
   assert slag_lower_waterwall_section
```

37. **PLANT_STATE_18**

```
IF the plant is operating at full load AND
   the control_system_stable_timer is zero AND
   this plant state has been in existence for a minimum of time
   determined by the PLANT_STATE_18 timer AND
   Reheat_Steam_Temp is in the low range AND
   Main_Steam_Temp is in the low range AND
   Gas_Exit_Temp is in the high range AND
   Burner_Tilt is in the high range AND
   Excess_Air is in the normal range AND
   Reheat_Steam_Spray is in the normal range AND
   Main_Steam_Spray is in the normal range
THEN
   assert slag_primary_section
   assert slag_secondary_section
   assert slag_middle_waterwall_section
```

38. **PLANT_STATE_19**

```
IF the plant is operating at full load AND
   the control_system_stable_timer is zero AND
   this plant state has been in existence for a minimum of time
   determined by the PLANT_STATE_19 timer AND
   Reheat_Steam_Temp is in the high range AND
   Main_Steam_Temp is in the normal range AND
   Gas_Exit_Temp is in the normal range AND
   Burner_Tilt is in the low range AND
   Excess_Air is in the normal range AND
   Reheat_Steam_Spray is in the normal range AND
   Main_Steam_Spray is in the normal range
THEN
   assert slag_middle_waterwall_section
```

39. **PLANT_STATE_20**

```
IF the plant is operating at full load AND
   the control_system_stable_timer is zero AND
   this plant state has been in existence for a minimum of time
   determined by the PLANT_STATE_20 timer AND
   Reheat_Steam_Temp is in the high range AND
   Main_Steam_Temp is in the normal range AND
   Gas_Exit_Temp is in the normal range AND
   Burner_Tilt is in the low range AND
   Excess_Air is in the normal range AND
   Reheat_Steam_Spray is in the normal range AND
   Main_Steam_Spray is in the normal range
THEN
   assert slag_primary_section AND
   assert slag_middle_waterwall_section
```

RULE TYPE : Boiler Configuration Rules

40. IF
    the fuel type is Minto_Coal AND
    the primary sootblowers have not been operated in a primary_cycle(8 hrs)
  THEN
    select a primary sootblowing strategy.

41. IF
    the fuel type is Minto_Coal AND
    the hot air_heaters have not been operated in a hot_air_heater cycle (24 hrs)
  THEN
    select a hot_air_heater sootblowing strategy.

42. IF
    the fuel type is Minto_Coal AND
    the cold air_heaters have not been operated in a cold_air_heater cycle (8 hrs)
  THEN
    select a cold_air_heater sootblowing strategy.

43. IF
    the fuel type is Minto_Coal AND
    the economizer sootblowers have not been operated in a economizer cycle (2 week)
  THEN
    select an economizer sootblowing strategy.

44. IF
    a mill is added or deleted from the boiler configuration
  THEN
    wait 10 mins for control system variables to reach a steady state before scanning for slagging conditions

APPENDIX IV

SOOTBLOWING PLANT DATA

DALHOUSIE UNIT 2 – OPERATING DATA

	ACTION 1	ACTION 2	ACTION 3	ACTION 4
BEFORE SOOTBLOW				
Shift	Shift 4–12	Shift 4–12	Shift 12–8	Shift 8–4
Date	01/24/90	01/24/90	01/25/90	01/25/90
Time	21:46	22:40		10:32
Main Steam Temp	1004			1009
Reheat Steam Temp	998			1004
Gas Exit Temp	680			690
Main Steam Pressure	1750			1800
Main Steam Spray	0			0
Reheat Steam Spray	0			0
Burner Tilt	30			30
Excess Air	2.4			2.3
AFTER SOOTBLOW				
Time	22:36			13:20
Main Steam Temp	990			992
Reheat Steam Temp	987			1005
Gas Exit Temp	656			680
Main Steam Pressure	1795			1800
Main Steam Spray	0			0
Reheat Steam Spray	0			0
Burner Tilt	75			50
Excess Air	2.5			2.3
FUEL CHARACTERISTICS				
Mill 1	Yes			Yes
Mill 2	Yes			Yes
Mill 3	Yes			Yes
Mill 4	Yes			Yes
SOOTBLOWING ACTION				
Middle Primary	IK37/38			
Final Reheat	IK27/28/29/30			
Hot Reheat				
Cold Reheat				IK14
Secondary	IK4/6			IK3
Upper Waterwall	IR37/40/43/46			
Middle Waterwall				IR13/15/18/21/22/24
Lower Waterwall			IR10/11/12	IR1/9/10/11/12
Slopes				IR63/67/68
Air Heater		IKAH1/2	IKAH1/2/3/4	IKAH1/2
Economizer				
Sootblowing Reasons	1			3/4(LWW,SBC)/5
BOILER OBSERVATIONS				
Top Boiler	Dust 1"		Dust 2.5" –Packed	Dust 1.5"
Primary	Dusty		Dust 1"	Dust 1"
Final	Dusty		Clean	Clean
Hot Reheat	Dusty		Clean	Clean
Cold Reheat			Clean	Clean
Secondary	Slag 6"		Slag 5"	Slag 4.5"
Upper Waterwall				Clean
Middle Waterwall				Clean
Lower Waterwall			West Wall Slag	Clean
Slope	B Corner Low		B Corner Low	B corner Low
ANALYSIS				
Reheat Steam Temp Diff	-14			-17
Main Steam Temp Diff	8			1
Gas Exit Temp Diff	-24			-10
Main Steam Pressure Diff	45			0
Main Steam Spray Diff	0			0
Reheat Steam Spray Diff	0			0
Excess Air Diff	0.1			0
Burner Tilt Diff	45			20

DALHOUSIE UNIT 2 – OPERATING DATA

ACTION 5	ACTION 6	ACTION 7	ACTION 8	ACTION 9
Shift 4–12	Shift 4–12	Shift 4–12	Shift 12–8	Shift 12–8
01/25/90	01/25/90	01/25/90	01/26/90	01/26/90
16:40	18:52	19:25	3:04	6:03
1000	998	990	1001	983
1008	1000	992	1001	984
681	676	678	683	683
1798	1795	1785	1802	1794
0	0	0	6.77	0
0	0	0	0	0
30	45	30	30	40
2.3	2.3	2.3	2.3	2.3
17:00	19:05	21:07	03:20	06:25
992	998	1001	993	988
985	1000	997	992	980
675	678	668	681	665
1796	1801	1795	1801	1792
0	0	0	0	0
0	0	0	0	0
50	30	95	50	40
2.3	2.3	2.3	2.3	2.3
Yes	Yes	Yes	Yes	Yes
Yes	Yes	Yes	Yes	Yes
Yes	Yes	Yes	Yes	Yes
Yes	Yes	Yes	Yes	Yes
		IK37/38/39		IK40/41/42
		IR44		
IR27/28/32		IR34/36/37	IR14/17/20/23	
	IR63/67/68			
		IKAH1/2		IKAH1/2/3/4
1/8	4	2/3/7	9/10	3/7
Dust 1.5"	Dust 1.5"	Dust 1.5"	Dust 1"	Dust 1"
Dusty	Dusty	Dusty	Dusty	Dusty
Dust 1"	Dust 1"	Dust 1"	Dusty	Dusty
Clean	Clean	Clean	Clean	Clean
Clean	Clean	Clean	Clean	Clean
Slag 3.5"	Slag 3.5"	Slag 3.5"	Slag 3"	Slag 3"
B Corner Low	B Corner Low	B Corner Low	B Corner Low	B Corner Low
-8	0	11	-8	-5
-13	0	6	-9	-4
-6	2	-10	-2	-18
-2	6	10	-1	8
0	0	0	-6.77	0
0	0	0	0	0
0	0	0	0	0
20	-15	35	20	0

DALHOUSIE UNIT 2 – OPERATING DATA

ACTION 10	ACTION 11	ACTION 12	ACTION 13	ACTION 14
Shift 8–4	Shift 4–12	Shift 4–12	Shift 12–8	Shift 8–4
01/26/90	01/26/90	01/26/90	01/27/90	01/27/90
	19:50	21:50	3:37	12:45
	998	994	1002	1004
	1004	991	1006	998
	674	678	679	677
	1808	1782	1794	1800
	0	0	0	0
	0	0	0	0
	35	35	36	33
	2.7	2.4	2.4	2.5
		22:58	5:40	13:07
	970	986	987	980
	981	990	1002	1002
	664	662	660	663
	1724	1794	1810	1778
	0	0	0	0
	0	0	0	0
	33	75	66	40
	2.8	2.4	2.5	2.6
	Yes	Yes	Yes	Yes
	Yes	Yes	Yes	Yes
	Yes	Yes	Yes	Yes
	Yes	Yes	Yes	Yes
		IK38/39/40	IK37/38	IK39/40/41
			IK25/26/31/32	IK27/28
		IK4/5		
	IR44/35	IR38/47		
IR23/19/25/29/28/30	IR26/29	IR27/32	IR28/29/31/33/35	
		IKAH1/2	IKAH1/2/3/4	
2	2	2/3/5/7	1/3/12	1/7
	Dust 1"	Dust 1"	Dust 1"	Dust 1.5"
	Dust 1.5"	Dust 1.5"	Clean	Clean
	Dust 1.5"	Dust 1.5"	Clean	Clean
	Clean	Clean	Clean	Clean
	Clean	Clean	Clean	clean
	Slag 5"	Slag 5"	Slag 2.5"	Slag 2.5"
				Slag
	B Corner Low	B Corner Low	B Corner Low	B Corner Low
	-29	-6	-5	-14
	-43	8	-3	3
	-8	-14	-18	-14
	-79	12	18	-22
	0	0	0	0
	0	0	0	0
	0.1	0	0.1	0
	0	40	30	7

DALHOUSIE UNIT 2 – OPERATING DATA

ACTION 15	ACTION 16	ACTION 17	ACTION 18	ACTION 19
Shift 8–4	Shift 4–12	Shift 4–12	Shift 12–8	Shift 8–4
01/27/90	01/27/90	01/27/90	01/28/90	01/28/90
14:36	19:56	20:50	5:30	10:08
1008	1006	1001	1007	984
1008	1009	998	995	1006
668	676	667	682	679
1780	1786	1785	1784	1790
0	0	0	0	0
0	0	0	0	0
26	30	56	30	42
2.5	2.3	2.3	2.5	2.4
14:45	20:10	21:03	6:25	10:40
1002	1000	996	999	985
1001	995	980	1001	1001
666	663	666	667	678
1798	1790	1792	1808	1800
0	0	0	0	0
0	0	0	0	0
26	56	66	70	82
2.6	2.3	2.3	2.4	2.4
Yes	Yes	Yes	Yes	Yes
Yes	Yes	Yes	Yes	Yes
Yes	Yes	Yes	Yes	Yes
Yes	Yes	Yes	Yes	Yes
			IK37/38/39	IK39/40
	IK26/27/28			IK30
IR38/44			IR38/41/44	IR41/47
IR24	IR28/32/34/36		IR26/29/32/35	IR28/30/34/38
		IR1/9/10/11/12		
	IR63/67/68	IR63/67/68		
		IKAH1/2	IKAH1/2/3/4	IKAH1/2
2	2/4/7	4(LWW,SSC)/3	3/7/8/9/12	3/7/8/9/12
Dust 1.5"	Dust 1.5"	Dust 1.5"	Dust 1.25"	Dust 1.5"
Clean	Dust 1"	Dust 1"	Dust 1"	Dust 1.5"
Clean			Dusty	Dust 1.5"
Clean			Dusty	Dust .5"
clean				
Slag 2.5"	Slag 2.5"	Slag 2.5"	Slag 3"	Slag 3"
Slag				
				Slag- West
B Corner Low	B Corner Low	B Corner Low	B Corner Low	B Corner Low
-6	-5	-3	-8	-1
-7	-5	-4	5	-4
-2	-16	-1	-15	-1
16	5	7	14	10
0	0	0	0	0
0	0	0	0	0
0	0	0	-0.1	0
0	25	10	40	20

DALHOUSIE UNIT 2 – OPERATING DATA

ACTION 20	ACTION 21	ACTION 22	ACTION 23	ACTION 24
Shift 4–12	Shift 4–12	Shift 4–12	Shift 4–12	Shift 12–8
01/28/90	01/28/90	01/28/90	01/28/90	01/29/90
15:46	17:11	20:12	23:05	1:15
1000	990	987	1010	1004
1010	1013	1002	1007	1000
675	664	667	677	680
1789	1797	1785	1800	1797
0	0	0	0	0
0	0	0	0	0
56	30	30	30	30
2.5	2.6	2.8	2.6	2.4
16:39	18:38	22:36		2:13
979	988	1001		1002
1006	995	998		1003
669	661	675		685
1802	1796	1801		1808
0	0	0		0
0	0	0		0
30	46	40		70
2.4	2.7	2.7		2.4
Yes	Yes	Yes		Yes
Yes	Yes	Yes		Yes
Yes	Yes	Yes		Yes
Yes	Yes	Yes		Yes
IK35/36/37	IK38/39/40			IK37/38/41/42
IK27/28	IK29/30			IK29/30
IK21/22				
IK3/4				IK5
		IR44		
	IR22/23/24/26	IR34/36	IR25/27/29	IR26/28/30/31/32
	IR1/9/10/11/12			
		IKAH1/2		
1	5	3/8	8/9	1
Dust 2"	Dust 2"	Dust 2"	Dust 2"	Dust 1"
Dust 1"	Dust 1"	Dust 1"	Dust 1"	Dusty
Dusty	Dusty	Dusty	Dusty	Dusty
Clean	Clean	Clean	Clean	Dusty
Clean	Clean	Clean	Clean	Dusty
Slag 2"	Slag 2"	Slag 2"	Slag 2"	Slag 6"
Slag–West	Slag–West	Slag–West	Slag–West	
Slag–West	Slag–West	Slag–West	Slag–West	
B/C Corner Low	B/C Corner Low	B/C Corner Low	B/C Corner Low	B Corner Low
-21	-1	4		-2
-6	-16	-4		3
-18	-3	8		-15
17	-2	16		11
0	0	0		0
0	6	0		0
-0.1	0.1	-0.2		0
-20	16	10		40

DALHOUSIE UNIT 2 – OPERATING DATA

ACTION 25	ACTION 26	ACTION 27	ACTION 29	ACTION 30
Shift 12–8	Shift 8–4	Shift 8–4	Shift 4–12	Shift 4–12
01/29/90	01/29/90	01/29/90	01/29/90	01/29/90
5:43	11:23	12:54	17:23	18:50
1003	998	972	1006	996
997	986	985	1007	998
677	685	680	679	678
1803	1790	1785	1801	1800
0	0	0	0	0
0	0	0	0	0
35	30	50	30	50
2.4	2.3	2.5	2.6	2.4
6:06	12:53	14:35	15:51	19:10
1002	972	992	995	1000
998	995	1005	995	1000
677	690	675	675	668
1800	1796	1799	1804	1785
0	0	0	0	0
0	0	0	0	0
60	60	60	60	60
2.4	2.3	2.3	2.6	2.4
Yes	Yes	Yes	Yes	Yes
Yes	Yes	Yes	Yes	Yes
Yes	Yes	Yes	Yes	Yes
Yes	Yes	Yes	Yes	Yes
		IK35/38/39/42		IK36/37/41
		IK26/27/30/31		
		IK20/21		
	IK11			
	IK8			
IR40/41				
IR34/35/36	IR14–16/18–20/34–36		IR26/28/30/32	
	IR1–12			
	IR63/67/68			
	IKAH1/2			IKAH1/2
12	3/4(LWW,SSC)/1/12/8	1/13	8/9	3/7
Dust 1"	Dust 1"	Dust 1"	Dust 0.75"	Dust 0.75"
Dusty	Dusty	Dust 0.5"	Dust 0.5"	Dust 0.5"
Dusty	Dusty	Dusty	Dusty	Dusty
Dusty	Dusty	Clean	Clean	Clean
Dusty	Dusty	Clean	Slag Low	Slag Low
Slag 6"	Slag 6"	Slag 2"	SLag 5"	SLag 5"
S Corner Low	S Corner Low	Clean		
+1	-26	20	-6	2
1	-3	120	-12	2
0	-6	-15	-3	-8
-3	5	4	3	-15
0	0	0	0	0
0	0	0	0	0
0	0	0	0	0
30	50	-20	30	10

DALHOUSIE UNIT 2 – OPERATING DATA

ACTION 33	ACTION 34	ACTION 35	ACTION 36	ACTION 37
Shift 4–12	Shift 12–8	Shift 8–4	Shift 4–12	Shift 4–12
01/29/90	01/30/90	01/30/90	01/30/90	01/30/90
22:11	3:28	11:58	16:56	20:40
1005	1000	987	1010	985
1004	996	992	1000	978
680	686	686	681	688
1790	1800	1777	1788	1780
0	8.5	10.7	0	0
0	0	0	0	0
30	30	40	35	55
2.6	2.3	2.8	2.4	2.4
22:26	4:07	12:36	17:10	20:56
1000	989	1001	990	986
998	972	988	970	1001
677	659	686	683	675
1790	1786	1786	1805	1782
0	0	0	0	0
0	0	0	0	0
35	60	42	50	56
2.5	2.3	2.4	2.5	2.4
Yes	Yes	Yes	Yes	Yes
Yes	Yes	Yes	Yes	Yes
Yes	Yes	Yes	Yes	Yes
Yes	Yes	Yes	Yes	Yes
	IK40/41/42			IK36/36/37
				IK10/11
IR22/23/24/25	IR13–24		IR21–28/30	
		IR1–12		
	IKAH1/2/3/4	IKAH1/2		IKAH1/2
1/5	3/7/8/9	3/10	8/9	13
Dust 0.75"	Dust 1" Packed	Dust 0.5"	Dust 0.5"	Dust 0.5"
Dust 0.5"	Dust 0.5"	Dust 0.5"	Dust 0.5"	Dust 0.5"
Dusty	Dusty	Dust 0.5"	Dust 0.5"	Dust 0.5"
Clean	Clean	Dusty	Dusty	Dusty
Slag Low	clean	Dusty	Dusty	Dusty
SLag 3"	Slag 6"	Slag 4"	Slag 4"	Slag 4"
	Slag B Corner			
	Clean	Slag B Corner	Slag B Corner	Slag B Corner
-6	-12	4	-30	3
-6	-26	-4	-30	23
-3	-17	0	-8	-15
-10	-15	19	22	2
0	-8.5	-10.7	0	0
0	0	0	0	0
-0.1	0	0.1	0.1	0
5	30	2	15	0

DALHOUSIE UNIT 2 – OPERATING DATA

ACTION 38	ACTION 39	ACTION 40	ACTION 41	ACTION 42
Shift 12–8	Shift 12–8	Shift 8–4	Shift 4–12	Shift 12–8
01/31/90	01/31/90	01/31/90	01/31/90	02/01/90
2:45	4:21			
998	993			
1003	998			
882	883			
1786	1788			
0	0			
0	0			
30	40			
2.4	2.4			
3:15	5:40			
986	982			
1000	996			
678	669			
1800	1800			
0	0			
0	0			
50	45			
2.5	2.3			
Yes	Yes			
Yes	Yes			
Yes	Yes			
Yes	Yes			
	IK35/38/39			
	IK30			
IR47				
IR32/35	IR14/17/20/23			
	IR1/11/12			
	IR63/67/68			
	IKAH/1/2/3/4	IKAH1/2	IKAH1/2	IKAH1/2/3/4
1/8	1/3/4/7	3	3	3
Dusty	Dusty			
Dusty	Dusty			
Dusty	Dusty			
Clean	Clean			
Slag 2"	Slag 2"			
Slag 3.5"	Slag 3.5"			
Slag B Corner	Slag B Corner			
-1	-1			
-3	0			
-3	-13			
14	12			
0	0			
0	0			
0.1	-0.1			
20	5			

DALHOUSIE UNIT 2 – OPERATING DATA

ACTION 43	ACTION 44	ACTION 45	ACTION 46	ACTION 47
Shift 8–4	Shift 4–12	Shift 12–8	Shift 8–4	Shift 4–12
02/01/90	02/01/90	02/02/90	02/02/90	02/02/90
			9:00	18:10
			1005	994
			1000	1002
			680	672
			1795	1773
			20.05	0
			0	0
			30	45
			3	2.5
			10:15	19:10
			980	996
			994	1001
			667	673
			1802	1791
			0	0
			0	0
			30	48
			2.6	2.5
			Yes	Yes
			Yes	Yes
			Yes	Yes
			Yes	Yes
			IK35–42	
	IK28/29		IK26–30	
	IK21/22		IK18–21	
	IK11/14			
			IR39/44	
			IR32/35/36	
				IR1/2/10–12
				IR63/67/68
IKAH1/2	IKAH1/2	IKAH1/2/3/4	IKAH1/2/3/4	
3	3	3	1/3/9/10	4
			Dust 2"	Dust 2.5"
			Dust 2"	Dust 0.5"
			Dust 2"	Clean
			Dust	Clean
			Clean	Clean
			Slag 2"	Slag 4"
				Slag - West
			Slag S Corner	Slag S Corner
			-25	1
			-6	-1
			-33	1
			7	18
			–20.05	0
			0	0
			–0.4	0
			0	3

DALHOUSIE UNIT 2 – OPERATING DATA

ACTION 48	ACTION 49	ACTION 50	ACTION 51	ACTION 52
Shift 4–12	Shift 12–8	Shift 8–4	Shift 4–12	Shift 12–8
02/02/90	02/03/90	02/03/90	02/03/90	02/04/90
	3:05		20:00	4:10
	994		995	1000
	1000		999	1005
	669		661	675
	1800		1798	1795
	0		0	0
	0		0	0
	35		75	35
	2.3		2.6	2.3
	3:20		21:30	5:40
	1004		999	1000
	1001		1000	1005
	662		644	670
	1806		1790	1790
	0		0	0
	0		0	0
	35		60	35
	2.3		2.5	2.3
	Yes		Yes	Yes
	Yes		Yes	Yes
	Yes		Yes	Yes
	Yes		Yes	Yes
IK35/36/37	IK 38/39/40	IK35–42	IK35–42	IK37/40/41
		IK25–32	IK26/27/30/31	IK28/29
IK20/21			IK17/20/21/24	
IK4/5				
			IR38/41/44/47	
IR23/34/38				IR17/22/23/24
		IR1–12		IR1/11/12
	IR63/67/68			IR63/67/68
IKAH1/2	IKAH1/2/3/4		IKAH1/2	IKAH1/2/3/4
		IK33/34		
3/7	3/4/7		3/5	3/4/7/12
	Dust 2"		Dust 1.5"	Dust 1"
	Dust 2"		Dust 0.5"	Dust 1.5"
	Clean		Dust 1"	Dust 1.5"
	Clean		Dusty	Dust 1"
	Clean		Dusty	Dust 1"
	Slag 4.5"		Slag 4"	Slag 5"
	Slag B Corner		Slag B corner	Slag B Corner
	10		4	0
	1		11	0
	-7		-17	-5
	6		5	-5
	0		0	0
	0		0	0
	0		-0.1	0
	-5		-15	3

DALHOUSIE UNIT 2 – OPERATING DATA

ACTION 53	ACTION 54	ACTION 55	ACTION 56	ACTION 57
Shift 8–4	Shift 4–12	Shift 4–12	Shift 12–8	Shit 8–4
02/04/90	02/04/90	02/04/90	02/05/90	02/05/90
10:00	14:15			7:30
1003	1009			1010
1004	1002			1010
668	672			780
1796	1800			1800
0	0			10
0	0			10
30	40			30
2.3	2.6			2.1
11:00	16:25			
997	1001			997
995	995			990
668	672			774
1795	1797			1800
0	0			0
0	0			0
45	40			55
2.1	2.4			2.1
Yes	Yes			Yes
Yes	Yes			Yes
Yes	Yes			Yes
Yes	Yes			Yes
IK37/38/39		IK40/41/42	IK35/36/37	
		IK28/29		
		IK20	IK21/22	
				IK13
			IK3/4	
IR37/40/43/46		IR38–44		
IR26/29/32/35	IR17/23/25/31		IR21–26	IR14/16/18/20/22
		IR1–12		IR1–12
IKAH1/2		IKAH1/2	IKAH1/2/3/4	IKAH1/2
3/7/12	9			1/3/4/8/9/10/11
Dust 1"	Dust 1"			Dust Packed
Dust 0.5"	Dust 0.5"			Dust 2"
Dust 2"	Dust 2"			
Dusty	Dusty			clean
Slag	Slag			Clean
Slag	Slag			Slag 5"
Slag B Corner	Slag B Corner			
-6	-8			-13
-6	-7			-12
0	0			-6
-1	-8			0
0	0			-10
0	0			-10
-0.2	-0.2			0
15	0			25

Sootblowing Reason Legend

1 – Gas Exit temp High
2 – Temperatures High
3 – Air Heater Maintenance
4 – Observation
5 – Balance Temperatures
6 – Economizer Maintenance
7 – Primary Maintenance
8 – Reheat Temp High
9 – Main temp High
10 – Eliminate Main spray
11 – Eliminate Reheat Spray
12 – Burner Tilt Low
13 – Temperatures Low

APPENDIX V

SOOTBLOWING TEST DATA AND RESULTS

APPENDIX V

SOOTBLOWING TEST DATA AND RESULTS

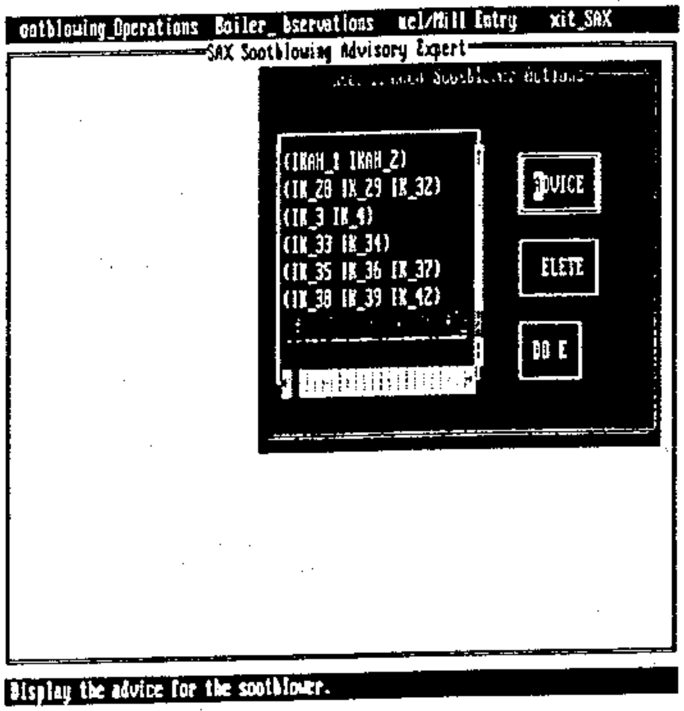
A SAX test consists of running the SAX system using the test data presented in this appendix. The results of these tests will be presented in the following way:

```
1.    The analysis of the test data and results
2.    The test data set at five minute intervals
before it has been linearly interpolated.
3.    For each plant state in this data which
requires a sootblowing action the sootblowing
strategy list and corresponding advice windows
will be presented.
```

It should be noted that there are discrepancies between the data collected by the operators and the data obtained from the strip charts for the test data sets. These discrepancies may have occurred for various reasons. It is possible that operators did not record the data as accurately or consistently as required. Inconsistencies in data test sets may have occurred for various reasons. The first problem is the determination of the position on the strip chart at which to start the readings. The only reference point available is a shift supervisor's initials which may occur every eight hours. Since the process variables are recorded on different strip charts these initials may not correctly represent the time recorded by the supervisor. As a result, each variable may be read at slightly different times. With this in mind the best possible test data sets were collected.

SAX TEST #1

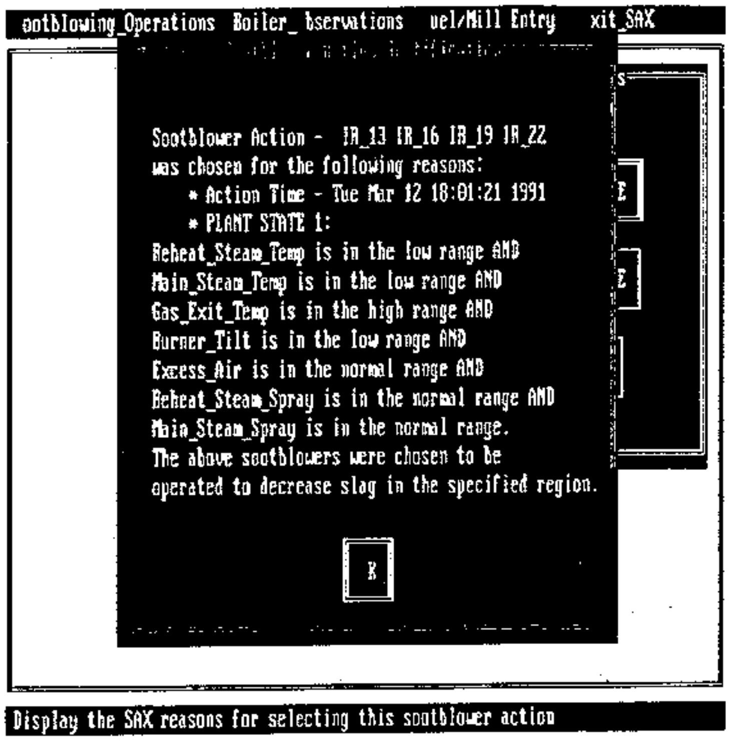
In this test data the reheat temperature began high but since there was no indication of a high gas exit temperature SAX took no action. As both main and reheat steam temperatures dropped and the gas exit temperature rose SAX detected a situation which required sootblowing, Plant State 1. As the burner tilt rose and the reheat and main steam temperatures still remained low, Plant State 8 was detected and further sootblowing strategies were chosen. The following pages contain the test data set and the SAX screens which display the results.



```
(IKAH_1 IKAH_2)
(IK_28 IK_29 IK_32)
(IK_3 IK_4)
(IK_33 IK_34)
(IK_35 IK_36 IK_37)
(IK_38 IK_39 IK_42)
```

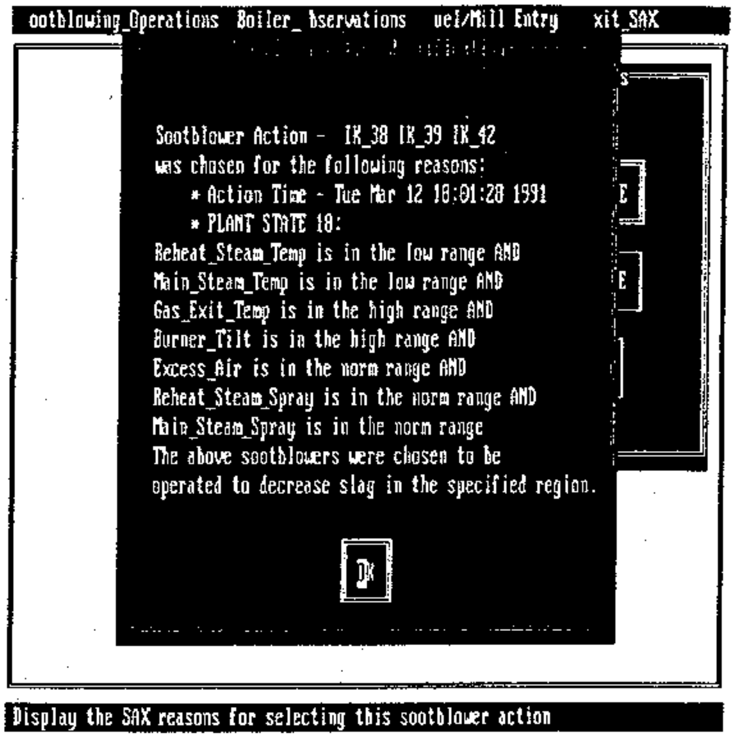
Display the advice for the sootblower.

Figure A5-1 SAX TEST #1 - Plant State 1 - Sootblowing Strategy List


ootblowing_Operations Boiler_ bservations uel/Mill Entry xit_SAX
Display the SAX reasons for selecting this sootblower action

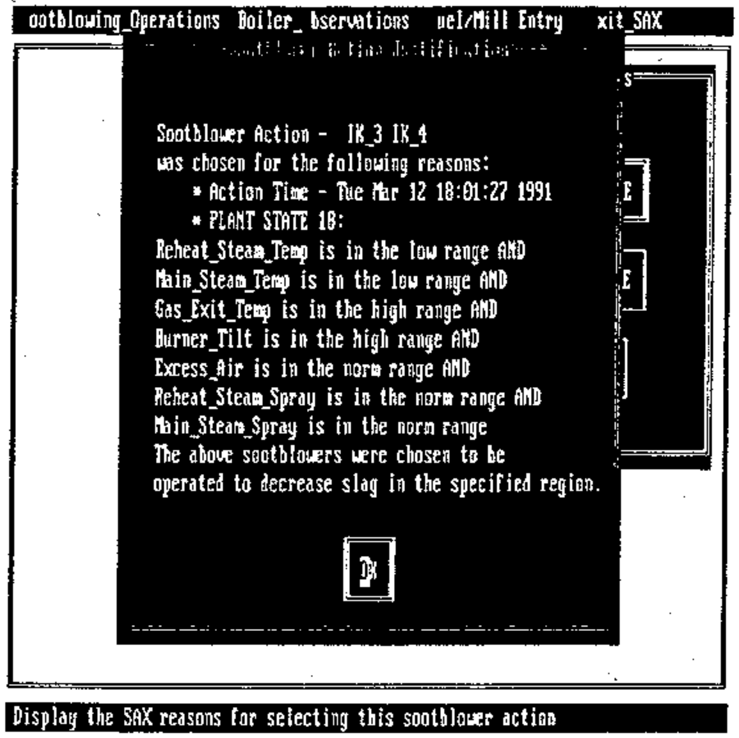
```
Sootblower Action -  IR_13 IR_16 IR_19 IR_22
was chosen for the following reasons:
    * Action Time - Tue Mar 12 18:01:21 1991
    * PLANT STATE 1:
Reheat_Steam_Temp is in the low range AND
Main_Steam_Temp is in the low range AND
Gas_Exit_Temp is in the high range AND
Burner_Tilt is in the low range AND
Excess_Air is in the normal range AND
Reheat_Steam_Spray is in the normal range AND
Main_Steam_Spray is in the normal range.
The above sootblowers were chosen to be
operated to decrease slag in the specified region.
```

Figure A5-2 SAX TEST #1 - Plant State 1 - Middle Waterwall Strategy


ootblowing_Operations Boiler_ bservations uel/Mill Entry xit_SAX
OK
Display the SAX reasons for selecting this sootblower action

```
Sootblower Action -  IK_38 IK_39 IK_42
was chosen for the following reasons:
    * Action Time - Tue Mar 12 18:01:28 1991
    * PLANT STATE 18:
Reheat_Steam_Temp is in the low range AND
Main_Steam_Temp is in the low range AND
Gas_Exit_Temp is in the high range AND
Burner_Tilt is in the high range AND
Excess_Air is in the norm range AND
Reheat_Steam_Spray is in the norm range AND
Main_Steam_Spray is in the norm range
The above sootblowers were chosen to be
operated to decrease slag in the specified region.
```

Figure A5-3 SAX TEST #1 - Plant State 18 - Primary Strategy


ootblowing_Operations Boiler_ bservations uel/Mill Entry xit_SAX
OK

```
Sootblower Action -  IK_3 IK_4
was chosen for the following reasons:
    * Action Time - Tue Mar 12 18:01:27 1991
    * PLANT STATE 18:
Reheat_Steam_Temp is in the low range AND
Main_Steam_Temp is in the low range AND
Gas_Exit_Temp is in the high range AND
Burner_Tilt is in the high range AND
Excess_Air is in the norm range AND
Reheat_Steam_Spray is in the norm range AND
Main_Steam_Spray is in the norm range
The above sootblowers were chosen to be
operated to decrease slag in the specified region.
```

```
Display the SAX reasons for selecting this sootblower action
```

Figure A5-4 SAX TEST #1 - Plant State 18 - Secondary Strategy

SAX TEST # 1

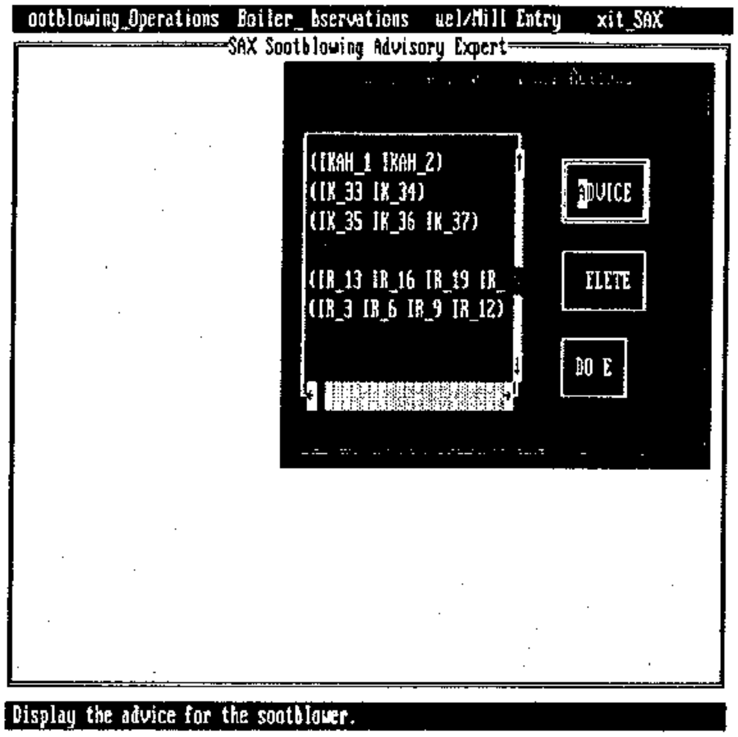
SOOTBLOWING ACTION 12

Process Variable	Units	Record	21:30	21:35	21:40	21:45	21:50	21:55
Main Steam Temperature	Fahrenheit	Strip Chart	850.00	1070.00	1070.00	940.00	900.00	950.00
Reheat Steam Temperature	Fahrenheit	Strip Chart	1000.00	950.00	850.00	850.00	880.00	880.00
Gas Exit Temperature	Fahrenheit	Hourly Log	660.00	665.00	667.00	670.00	680.00	685.00
Main Steam Pressure	PSIG	Strip Chart	1790.00	1790.00	1790.00	1790.00	1795.00	1795.00
Excess O2	%	Strip chart	2.40	2.30	2.40	2.70	2.30	2.60
Steam Flow	lbs/hr x 10^4	Strip Chart	136.00	140.00	135.00	135.00	137.00	137.00
Burner Tilt	Degrees	Trend Log	35.00	35.00	35.00	35.00	60.00	60.00
Main Steam Attemporating Spray	lbs/hr x 10^3	Hourly Log	0.00	0.00	0.00	0.00	0.00	0.00
Reheat Steam Attemporating Spray	lbs/hr x 10^3	Hourly Log	0.00	0.00	0.00	0.00	0.00	0.00

Process Variable	22:00	22:05	22:10	22:15	22:20	22:25	22:30
Main Steam Temperature	945.00	950.00	800.00	620.00	700.00	890.00	900.00
Reheat Steam Temperature	900.00	750.00	500.00	600.00	860.00	900.00	850.00
Gas Exit Temperature	680.00	675.00	670.00	660.00	655.00	652.00	650.00
Main Steam Pressure	1796.00	1795.00	1800.00	1855.00	1805.00	1800.00	1800.00
Excess O2	2.40	2.40	2.20	2.60	2.30	1.70	2.40
Steam Flow	138.00	138.00	135.00	138.00	135.00	135.00	138.00
Burner Tilt	60.00	60.00	75.00	75.00	75.00	75.00	75.00
Main Steam Attemporating Spray	0.00	0.00	0.00	0.00	0.00	0.00	0.00
Reheat Steam Attemporating Spray	0.00	0.00	0.00	0.00	0.00	0.00	0.00

SAX Test #2

In this test SAX detected the slagging condition, Plant State 8 in which the main steam temperature and gas exit temperatures were high. The excess air was also high. SAX recommended a primary strategy for this situation. SAX also detected a change in excess air and matched on Plant State 4. This resulted in the selection of lower and middle waterwall strategies.

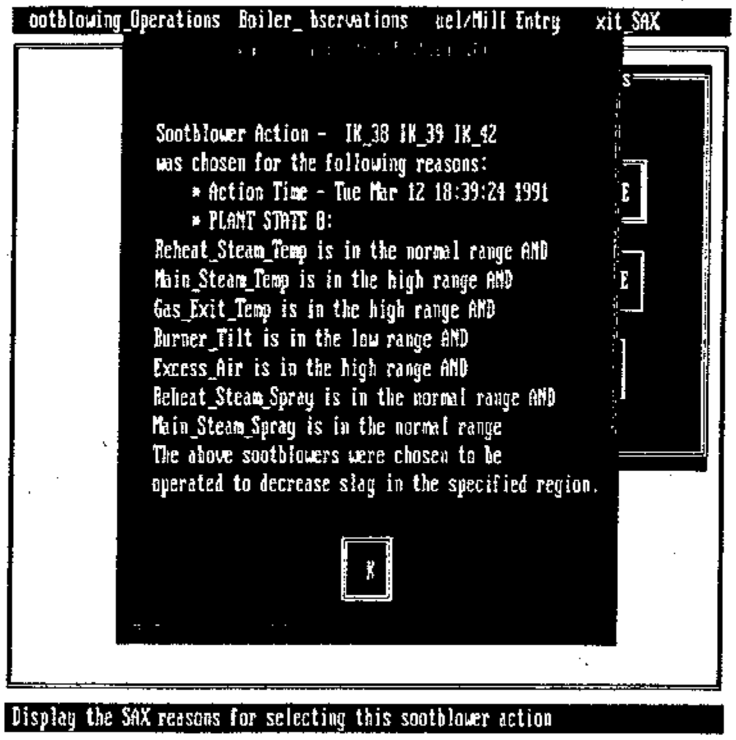


```
(IKAH_1 IKAH_2)
(IK_33 IK_34)
(IK_35 IK_36 IK_37)

(IR_13 IR_16 IR_19 IR_
(IR_3 IR_6 IR_9 IR_12)
```

Display the advice for the sootblower.

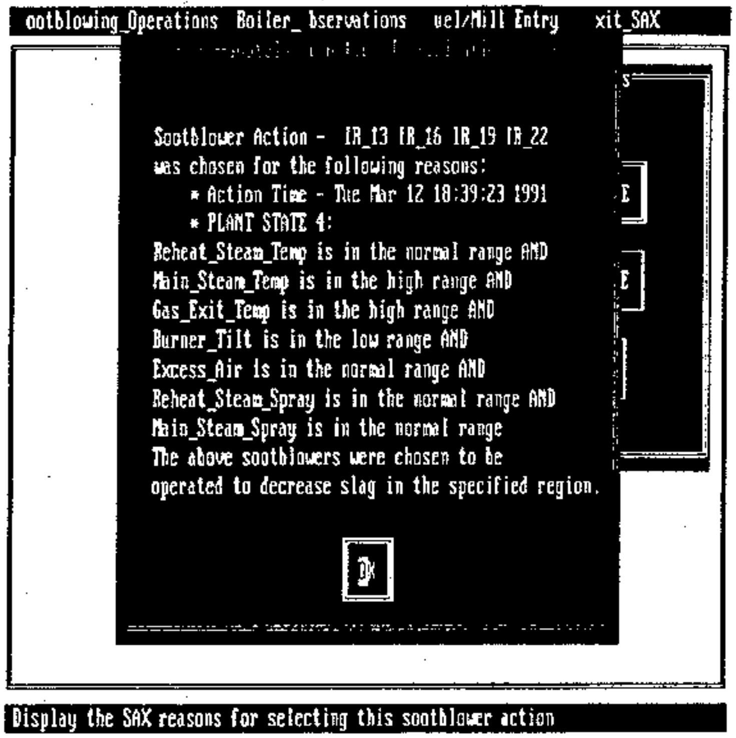
Figure A5-5 SAX TEST #2 - Plant State 8 - Sootblowing Strategy List


ootblowing_Operations Boiler_ bservations uel/Mill Entry xit_SAX

```
Sootblower Action -  IK_38 IK_39 IK_42
was chosen for the following reasons:
    * Action Time - Tue Mar 12 18:39:24 1991
    * PLANT STATE 8:
Reheat_Steam_Temp is in the normal range AND
Main_Steam_Temp is in the high range AND
Gas_Exit_Temp is in the high range AND
Burner_Tilt is in the low range AND
Excess_Air is in the high range AND
Reheat_Steam_Spray is in the normal range AND
Main_Steam_Spray is in the normal range
The above sootblowers were chosen to be
operated to decrease slag in the specified region.
```

Display the SAX reasons for selecting this sootblower action

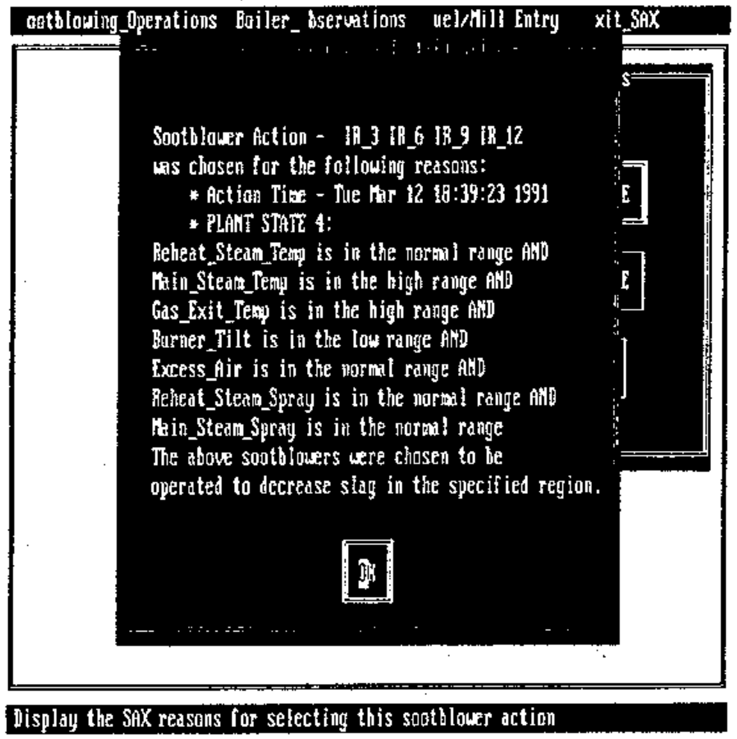
Figure A5-6 SAX TEST #2 - Plant State 8 - Primary Strategy


ootblowing_Operations Boiler_ bservations uel/Mill Entry xit_SAX

```
Sootblower Action -  IR_13 IR_16 IR_19 IR_22
was chosen for the following reasons:
    * Action Time - Tue Mar 12 18:39:23 1991
    * PLANT STATE 4:
Reheat_Steam_Temp is in the normal range AND
Main_Steam_Temp is in the high range AND
Gas_Exit_Temp is in the high range AND
Burner_Tilt is in the low range AND
Excess_Air is in the normal range AND
Reheat_Steam_Spray is in the normal range AND
Main_Steam_Spray is in the normal range
The above sootblowers were chosen to be
operated to decrease slag in the specified region.
```

OK
Display the SAX reasons for selecting this sootblower action

Figure A5-7 SAX TEST #2 - Plant State 4 - Middle Waterwall Strategy


ootblowing_Operations Boiler_ bservations uel/Mill Entry xit_SAX
OK

```
Sootblower Action -  IR_3 IR_6 IR_9 IR_12
was chosen for the following reasons:
    * Action Time - Tue Mar 12 18:39:23 1991
    * PLANT STATE 4:
Reheat_Steam_Temp is in the normal range AND
Main_Steam_Temp is in the high range AND
Gas_Exit_Temp is in the high range AND
Burner_Tilt is in the low range AND
Excess_Air is in the normal range AND
Reheat_Steam_Spray is in the normal range AND
Main_Steam_Spray is in the normal range
The above sootblowers were chosen to be
operated to decrease slag in the specified region.
```

Display the SAX reasons for selecting this sootblower action

Figure A5-8 SAX TEST #2 - Plant State 4 - Lower Waterwall Strategy

SAX TEST # 2

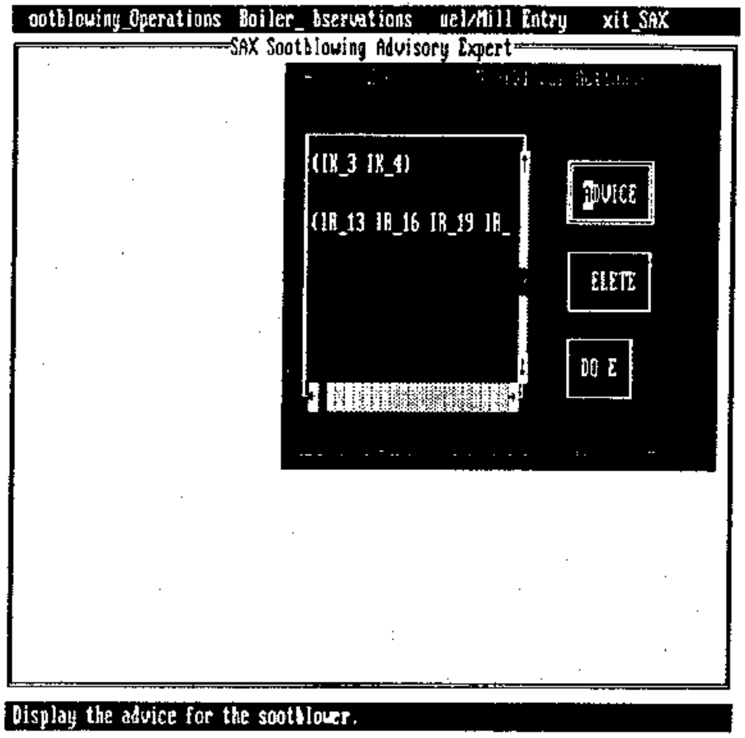
SOOTBLOWING ACTION 22

Process Variable	Units	Record	19:30	19:35	19:40	19:45	19:50	19:55
Main Steam Temperature	Fahrenheit	Strip Chart	1025.00	1005.00	1000.00	998.00	1000.00	1050.00
Reheat Steam Temperature	Fahrenheit	Strip Chart	990.00	980.00	950.00	980.00	1000.00	1000.00
Gas Exit Temperature	Fahrenheit	Hourly Log	660.00	660.00	660.00	665.00	665.00	670.00
Main Steam Pressure	PSIG	Strip Chart	1795.00	1795.00	1795.00	1795.00	1795.00	1795.00
Excess O2	%	Strip chart	2.60	2.60	2.80	2.40	2.40	2.40
Steam Flow	lbs/hr x 10^4	Strip Chart	137.00	135.00	135.00	135.00	135.00	136.00
Burner Tilt	Degrees	Trend Log	30.00	30.00	30.00	30.00	30.00	30.00
Main Steam Attemporating Spray	lbs/hr x 10^3	Hourly Log	0.00	0.00	0.00	0.00	0.00	0.00
Reheat Steam Attemporating Spray	lbs/hr x 10^3	Hourly Log	0.00	0.00	0.00	0.00	0.00	0.00

Process Variable	20:00	20:05	20:10	20:15	20:20	20:25	20:30
Main Steam Temperature	1045.00	1045.00	1050.00	1045.00	1020.00	1030.00	900.00
Reheat Steam Temperature	1000.00	1005.00	1000.00	990.00	940.00	750.00	610.00
Gas Exit Temperature	669.00	667.00	667.00	670.00	672.00	674.00	675.00
Main Steam Pressure	1800.00	1800.00	1800.00	1800.00	1800.00	1800.00	1800.00
Excess O2	2.40	2.40	2.40	2.80	2.60	2.80	3.00
Steam Flow	136.00	136.00	136.00	136.00	135.00	135.00	135.00
Burner Tilt	30.00	30.00	30.00	40.00	40.00	40.00	40.00
Main Steam Attemporating Spray	0.00	0.00	0.00	0.00	0.00	0.00	0.00
Reheat Steam Attemporating Spray	0.00	0.00	0.00	0.00	0.00	0.00	0.00

SAX Test #3

In this test data set SAX detected Plant State 18 in which the reheat and main steam temperatures were low and the burner tilt was high. The gas exit temperature was also high indicating a loss of heat in the boiler. SAX responded by recommending sootblowing actions in the primary, secondary and middle waterwall boiler sections.

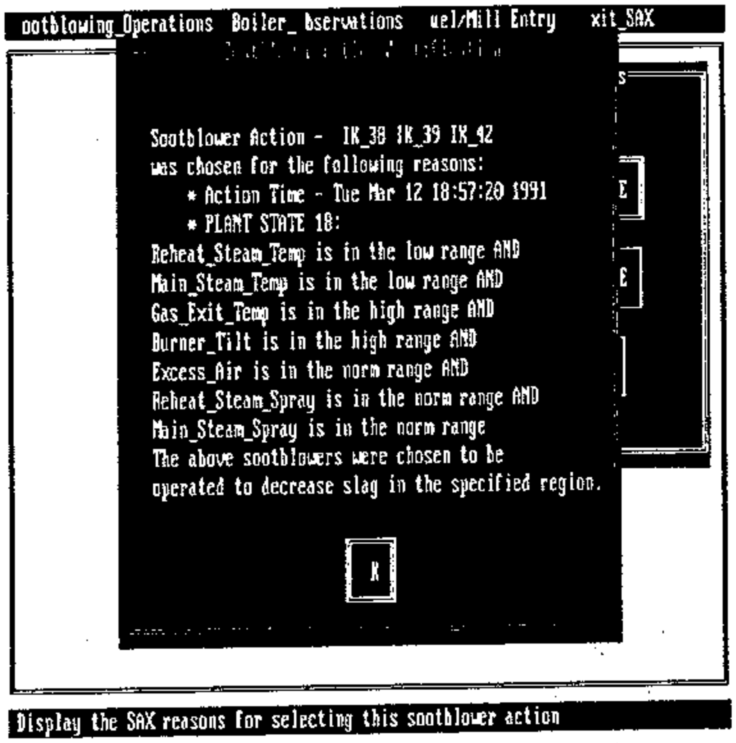

ootblowing_Operations Boiler_ bservations uel/Mill Entry xit_SAX
SAX Sootblowing Advisory Expert
DVICE
ELETE
DO E

(IK_3 IK_4)

(IR_13 IR_16 IR_19 IR_

Display the advice for the sootblower.

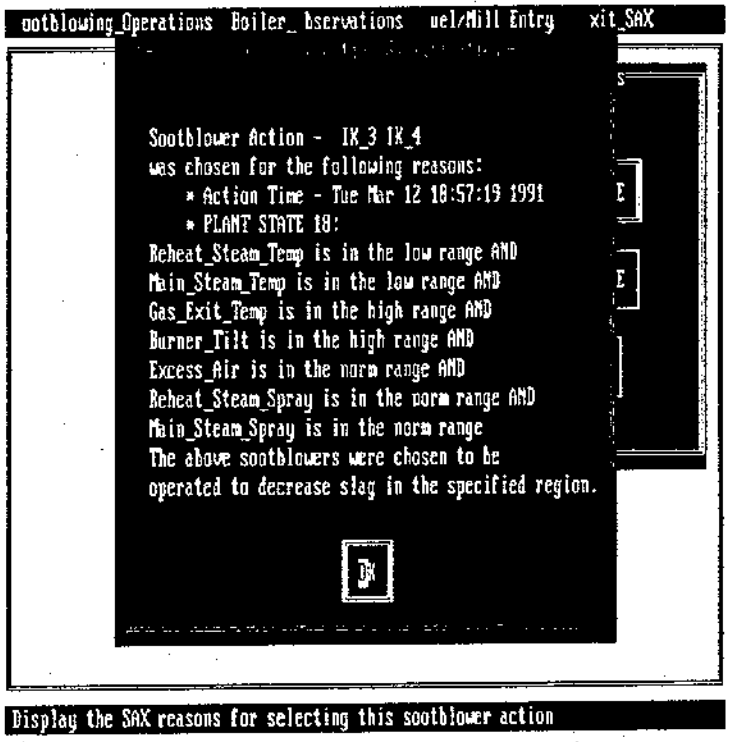
Figure A5-9 SAX TEST #3 - Plant State 18 - Sootblowing Strategy List


ootblowing_Operations Boiler_ bservations uel/Mill Entry xit_SAX

```
Sootblower Action -  IK_38 IK_39 IK_42
was chosen for the following reasons:
    * Action Time - Tue Mar 12 18:57:20 1991
    * PLANT STATE 18:
Reheat_Steam_Temp is in the low range AND
Main_Steam_Temp is in the low range AND
Gas_Exit_Temp is in the high range AND
Burner_Tilt is in the high range AND
Excess_Air is in the norm range AND
Reheat_Steam_Spray is in the norm range AND
Main_Steam_Spray is in the norm range
The above sootblowers were chosen to be
operated to decrease slag in the specified region.
```

Display the SAX reasons for selecting this sootblower action

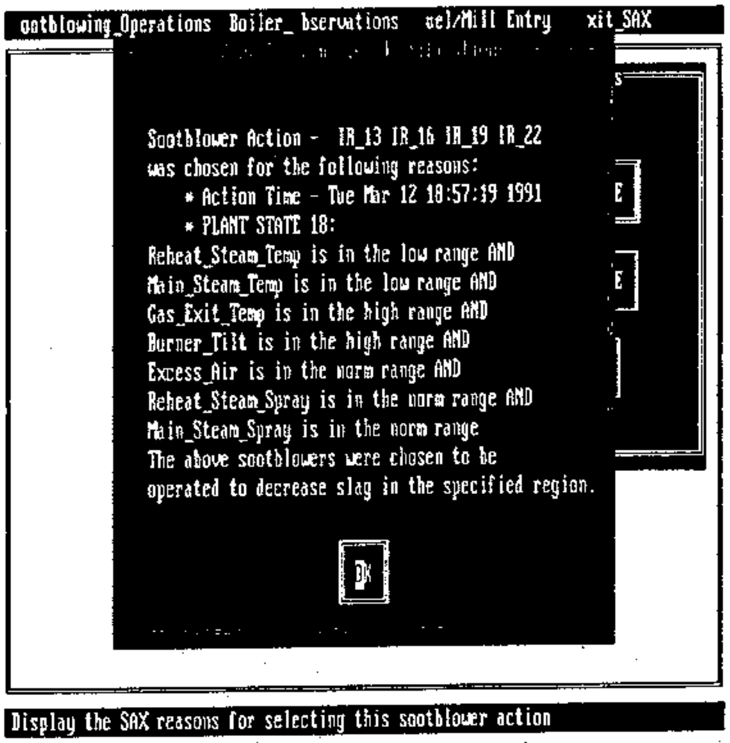
Figure A5-10 SAX TEST #3 - Plant State 18 - Primary Strategy


ootblowing_Operations Boiler_ bservations uel/Mill Entry xit_SAX

```
Sootblower Action -  IK_3 IK_4
was chosen for the following reasons:
    * Action Time - Tue Mar 12 18:57:19 1991
    * PLANT STATE 18:
Reheat_Steam_Temp is in the low range AND
Main_Steam_Temp is in the low range AND
Gas_Exit_Temp is in the high range AND
Burner_Tilt is in the high range AND
Excess_Air is in the norm range AND
Reheat_Steam_Spray is in the norm range AND
Main_Steam_Spray is in the norm range
The above sootblowers were chosen to be
operated to decrease slag in the specified region.
```

Display the SAX reasons for selecting this sootblower action

Figure A5-11 SAX TEST #3 - Plant State 18 - Secondary Strategy


ootblowing_Operations Boiler_ bservations uel/Mill Entry xit_SAX

Sootblower Action - IR_13 IR_16 IR_19 IR_22
was chosen for the following reasons:
* Action Time - Tue Mar 12 18:57:19 1991
* PLANT STATE 18:
Reheat_Steam_Temp is in the low range AND
Main_Steam_Temp is in the low range AND
Gas_Exit_Temp is in the high range AND
Burner_Tilt is in the high range AND
Excess_Air is in the norm range AND
Reheat_Steam_Spray is in the norm range AND
Main_Steam_Spray is in the norm range
The above sootblowers were chosen to be
operated to decrease slag in the specified region.

Display the SAX reasons for selecting this sootblower action

Figure A5-12 SAX TEST #3 - Plant State 18 - Middle Waterwall Strategy

SAX TEST # 3

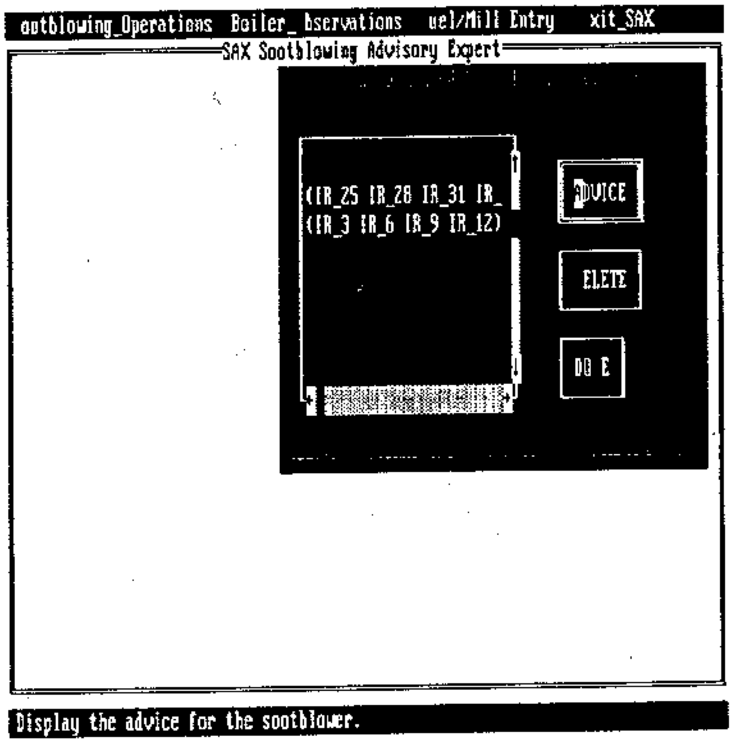
SOOTBLOWING ACTION 27

Process Variable	Units	Record	12:30	12:35	12:40	12:45	12:50	12:55
Main Steam Temperature	Fahrenheit	Strip Chart	705.00	710.00	780.00	795.00	790.00	780.00
Reheat Steam Temperature	Fahrenheit	Strip Chart	750.00	800.00	800.00	800.00	800.00	810.00
Gas Exit Temperature	Fahrenheit	Hourly Log	680.00	680.00	680.00	680.00	680.00	690.00
Main Steam Pressure	PSIG	Strip Chart	1794.00	1794.00	1794.00	1794.00	1794.00	1794.00
Excess O2	%	Strip chart	2.10	2.10	2.10	2.10	2.10	2.10
Steam Flow	lbs/hr x 10^4	Strip Chart	136.00	136.00	137.00	137.00	137.00	136.00
Burner Tilt	Degrees	Trend Log	80.00	80.00	80.00	80.00	80.00	80.00
Main Steam Attemporating Spray	lbs/hr x 10^3	Hourly Log	0.00	0.00	0.00	0.00	0.00	0.00
Reheat Steam Attemporating Spray	lbs/hr x 10^3	Hourly Log	0.00	0.00	0.00	0.00	0.00	0.00

Process Variable	13:00	13:05	13:10	13:15	13:20	13:25	13:30
Main Steam Temperature	800.00	805.00	805.00	890.00	895.00	840.00	800.00
Reheat Steam Temperature	810.00	900.00	900.00	900.00	910.00	810.00	710.00
Gas Exit Temperature	685.00	685.00	682.00	680.00	676.00	675.00	675.00
Main Steam Pressure	1798.00	1798.00	1798.00	1798.00	1798.00	1798.00	1798.00
Excess O2	1.70	2.30	1.80	2.40	2.00	2.20	2.60
Steam Flow	136.00	137.00	137.00	137.00	136.00	136.00	136.00
Burner Tilt	60.00	60.00	60.00	60.00	60.00	60.00	60.00
Main Steam Attemporating Spray	0.00	0.00	0.00	0.00	0.00	0.00	0.00
Reheat Steam Attemporating Spray	0.00	0.00	0.00	0.00	0.00	0.00	0.00

SAX Test #4

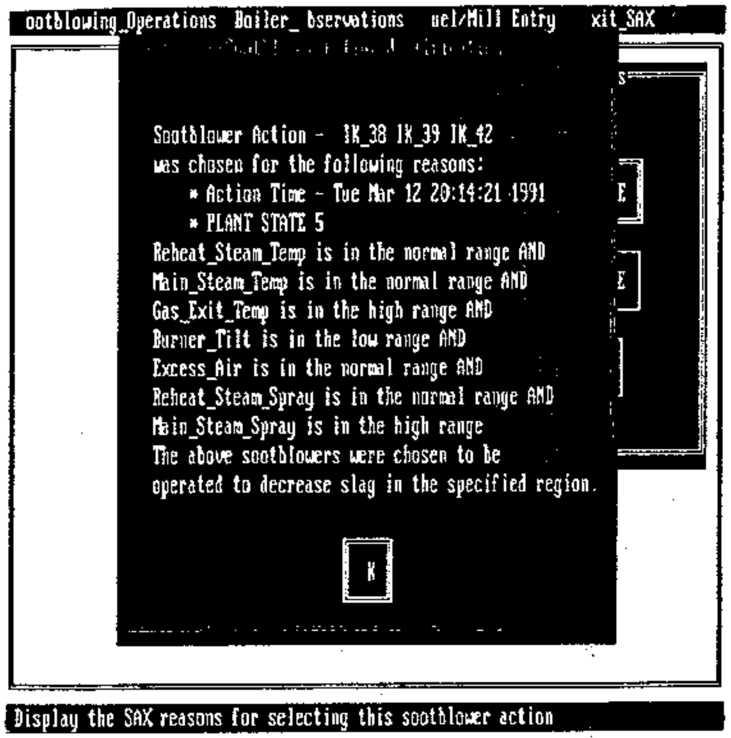
The data in this test set triggered on various plant states. Plant States 9, 1 and 5 were detected. Plant State 5 triggered since main steam attemporating spray was being used to control the main steam temperature. SAX recommended sootblowing in the primary, middle and lower waterwall sections of the boiler. The screens for plant states 1 and 9 will not be shown here, however plant state 5 screens are on the following pages.


ootblowing_Operations Boiler_ bservations uel/Mill Entry xit_SAX
SAX Sootblowing Advisory Expert
ADVICE
ELETE
DO E

(IR_25 IR_28 IR_31 IR_
(IR_3 IR_6 IR_9 IR_12)

Display the advice for the sootblower.

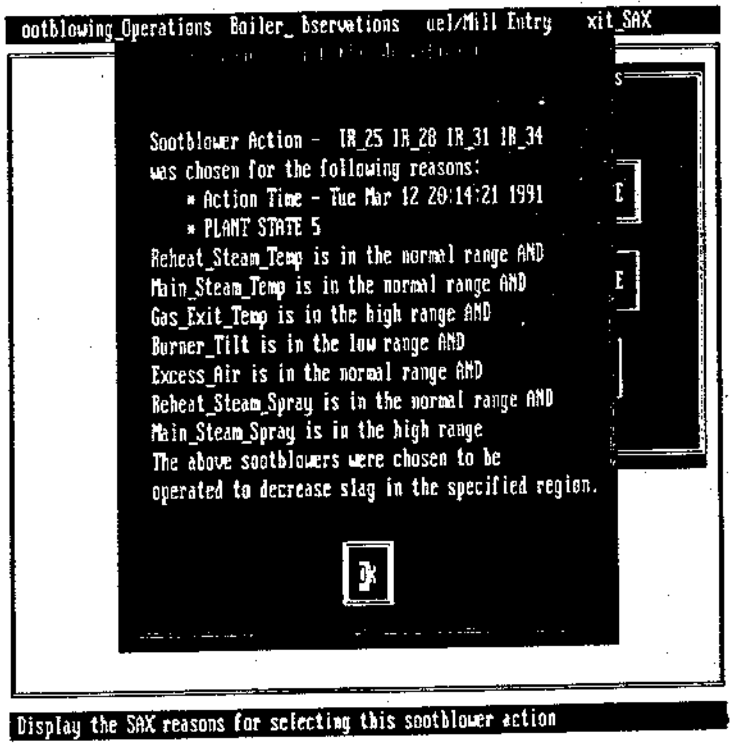
Figure A5-13 SAX TEST #4 - Plant State 5 - Sootblowing Strategy List


ootblowing_Operations Boiler_ bservations uel/Mill Entry xit_SAX

```
Sootblower Action -  IK_38 IK_39 IK_42
was chosen for the following reasons:
    * Action Time - Tue Mar 12 20:14:21 1991
    * PLANT STATE 5
Reheat_Steam_Temp is in the normal range AND
Main_Steam_Temp is in the normal range AND
Gas_Exit_Temp is in the high range AND
Burner_Tilt is in the low range AND
Excess_Air is in the normal range AND
Reheat_Steam_Spray is in the normal range AND
Main_Steam_Spray is in the high range
The above sootblowers were chosen to be
operated to decrease slag in the specified region.
```

Display the SAX reasons for selecting this sootblower action

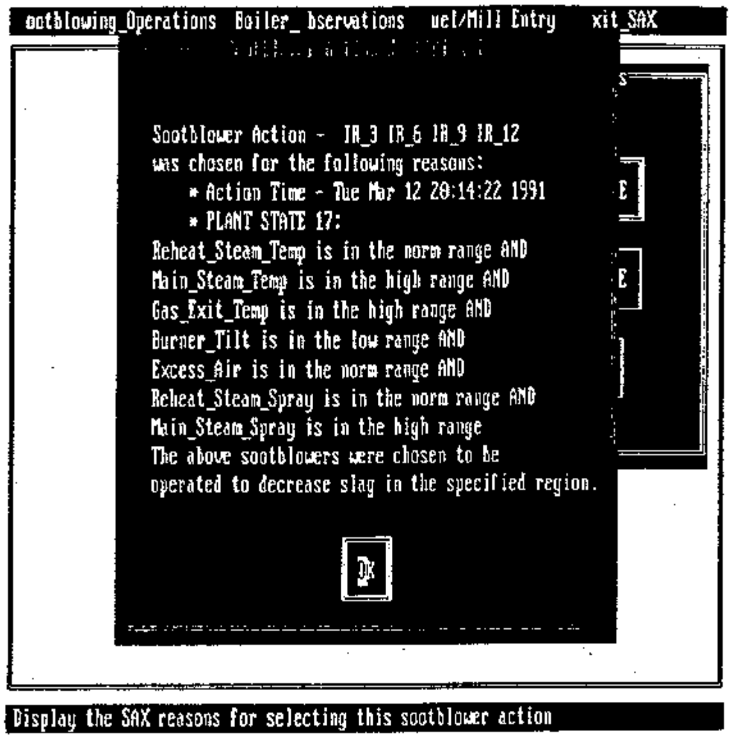
Figure A5-14 SAX TEST #4 - Plant State 5 - Primary Strategy


ootblowing_Operations Boiler_ bservations uel/Mill Entry xit_SAX

```
Sootblower Action -  IR_25 IR_28 IR_31 IR_34
was chosen for the following reasons:
    * Action Time - Tue Mar 12 20:14:21 1991
    * PLANT STATE 5
Reheat_Steam_Temp is in the normal range AND
Main_Steam_Temp is in the normal range AND
Gas_Exit_Temp is in the high range AND
Burner_Tilt is in the low range AND
Excess_Air is in the normal range AND
Reheat_Steam_Spray is in the normal range AND
Main_Steam_Spray is in the high range
The above sootblowers were chosen to be
operated to decrease slag in the specified region.
```

Display the SAX reasons for selecting this sootblower action

Figure A5-15 SAX TEST #4 - Plant State 5 - Middle Waterwall Strategy


ootblowing_Operations Boiler_ bservations uel/Mill Entry xit_SAX
OK
Display the SAX reasons for selecting this sootblower action

```
Sootblower Action - IR_3 IR_6 IR_9 IR_12
was chosen for the following reasons:
    * Action Time - Tue Mar 12 20:14:22 1991
    * PLANT STATE 17:
Reheat_Steam_Temp is in the norm range AND
Main_Steam_Temp is in the high range AND
Gas_Exit_Temp is in the high range AND
Burner_Tilt is in the low range AND
Excess_Air is in the norm range AND
Reheat_Steam_Spray is in the norm range AND
Main_Steam_Spray is in the high range
The above sootblowers were chosen to be
operated to decrease slag in the specified region.
```

Figure A5-16 SAX TEST #4 - Plant State 17 - Lower Waterwall Strategy

SAX TEST # 4

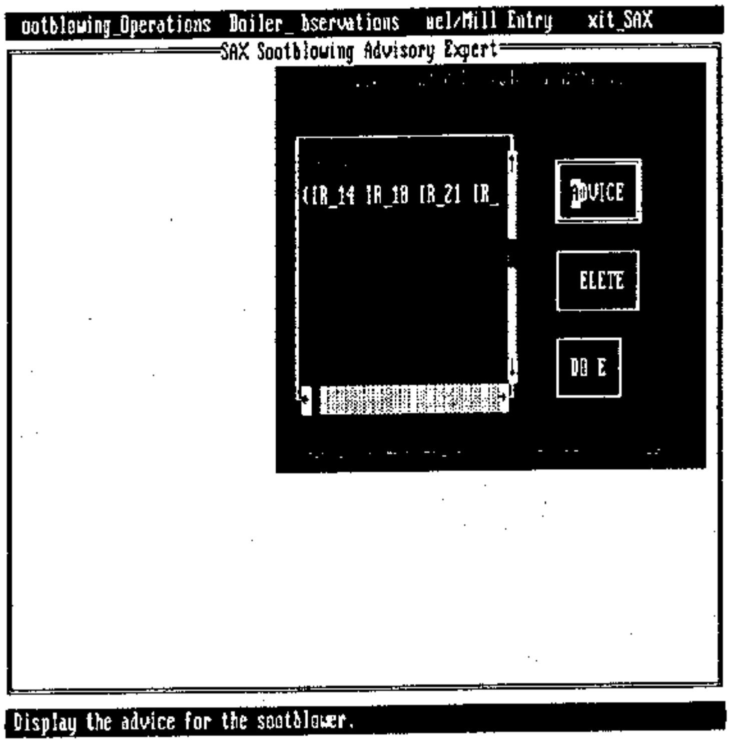
SOOTBLOWING ACTION 46

Process Variable	Units	Record	8:30	8:35	8:40	8:45	8:50	8:55
Main Steam Temperature	Fahrenheit	Strip Chart	990.00	970.00	945.00	990.00	1000.00	995.00
Reheat Steam Temperature	Fahrenheit	Strip Chart	900.00	880.00	920.00	910.00	910.00	1000.00
Gas Exit Temperature	Fahrenheit	Hourly Log	696.00	690.00	687.00	688.00	689.00	690.00
Main Steam Pressure	PSIG	Strip Chart	1851.00	1851.00	1851.00	1851.00	1851.00	1851.00
Excess O2	%	Strip chart	2.30	2.00	2.20	2.40	2.50	2.50
Steam Flow	lbs/hr x 10^4	Strip Chart	136.00	136.00	137.00	138.00	135.00	137.00
Burner Tilt	Degrees	Trend Log	30.00	30.00	30.00	30.00	30.00	30.00
Main Steam Attemporating Spray	lbs/hr x 10^3	Hourly Log	0.00	0.00	0.00	0.00	0.00	0.00
Reheat Steam Attemporating Spray	lbs/hr x 10^3	Hourly Log	0.00	0.00	0.00	0.00	0.00	20.00

Process Variable	9:00	9:05	9:10	9:15	9:20	9:25	9:30
Main Steam Temperature	940.00	990.00	830.00	790.00	800.00	810.00	815.00
Reheat Steam Temperature	980.00	950.00	920.00	975.00	800.00	900.00	810.00
Gas Exit Temperature	683.00	680.00	670.00	665.00	660.00	660.00	657.00
Main Steam Pressure	1799.00	1799.00	1799.00	1799.00	1799.00	1799.00	1799.00
Excess O2	2.30	2.30	2.30	2.30	2.40	2.40	2.40
Steam Flow	135.00	135.00	137.00	136.00	136.00	136.00	136.00
Burner Tilt	30.00	30.00	30.00	30.00	30.00	30.00	30.00
Main Steam Attemporating Spray	0.00	0.00	0.00	0.00	0.00	0.00	0.00
Reheat Steam Attemporating Spray	20.00	20.00	0.00	0.00	0.00	0.00	0.00

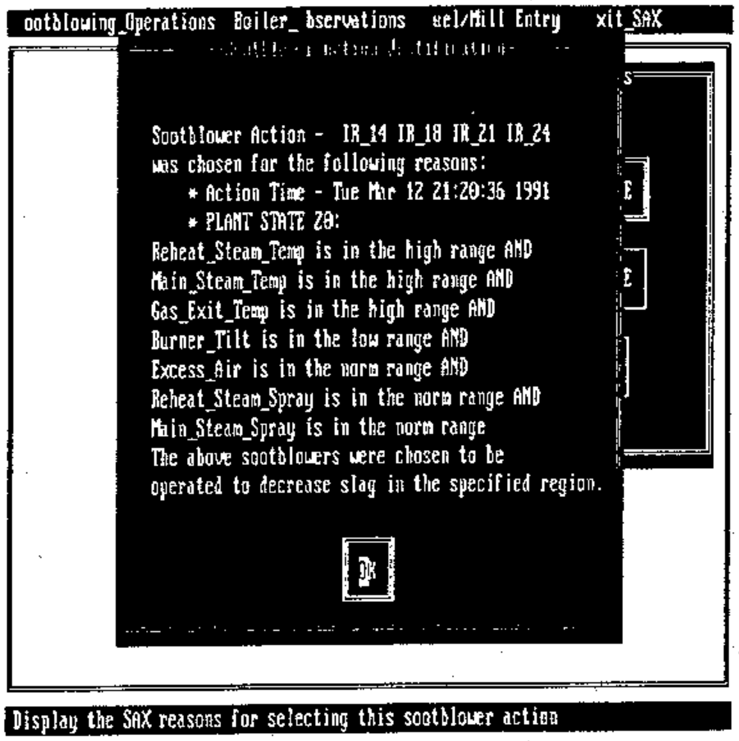
SAX Test #5

The data in this test set trigged on Plant State 1 in which the main and reheat temperatures were low and the gas exit temperature was high. Two sootblowing strategies were chosen. They were in the final reheat area and the middle waterwall area of the boiler. Later in the test set Plant State 20 was detected as a result of reheat steam, main steam and gas exit temperatures being high. In this plant state the sootblowing strategies for the primary region and the middle waterwall region were recommended by SAX. The Plant State 20 screens are presented on the following pages.


ootblowing_Operations Boiler_ bservations uel/Mill Entry xit_SAX
SAX Sootblowing Advisory Expert
(IR_14 IR_18 IR_21 IR_
DVICE
ELETE
DO E

Display the advice for the sootblower.

Figure A5-17 SAX TEST #5 - Plant State 20 - Sootblowing Strategy List



ootblowing_Operations Boiler_ bservations uel/Mill Entry xit_SAX

```
Sootblower Action -  IR_14 IR_18 IR_21 IR_24
was chosen for the following reasons:
    * Action Time - Tue Mar 12 21:20:36 1991
    * PLANT STATE 20:
Reheat_Steam_Temp is in the high range AND
Main_Steam_Temp is in the high range AND
Gas_Exit_Temp is in the high range AND
Burner_Tilt is in the low range AND
Excess_Air is in the norm range AND
Reheat_Steam_Spray is in the norm range AND
Main_Steam_Spray is in the norm range
The above sootblowers were chosen to be
operated to decrease slag in the specified region.
```

OK
Display the SAX reasons for selecting this sootblower action

Figure A5-18 SAX TEST #5 - Plant State 20 - Primary Strategy

SAX TEST # 5

SOOTBLOWING ACTION 49

Process Variable	Units	Record	2:30	2:35	2:40	2:45	2:50	2:55
Main Steam Temperature	Fahrenheit	Strip Chart	905.00	950.00	900.00	900.00	980.00	915.00
Reheat Steam Temperature	Fahrenheit	Strip Chart	950.00	900.00	950.00	950.00	945.00	950.00
Gas Exit Temperature	Fahrenheit	Hourly Log	660.00	665.00	665.00	665.00	668.00	669.00
Main Steam Pressure	PSIG	Strip Chart	1801.00	1801.00	1801.00	1801.00	1801.00	1801.00
Excess O2	%	Strip chart	2.30	2.30	2.30	2.30	2.30	2.30
Steam Flow	lbs/hr x 10^4	Strip Chart	134.00	134.00	135.00	134.00	134.00	134.00
Burner Tilt	Degrees	Trend Log	35.00	35.00	35.00	35.00	35.00	35.00
Main Steam Attemporating Spray	lbs/hr x 10^3	Hourly Log	0.00	0.00	0.00	0.00	0.00	0.00
Reheat Steam Attemporating Spray	lbs/hr x 10^3	Hourly Log	0.00	0.00	0.00	0.00	0.00	0.00

Process Variable	3:00	3:05	3:10	3:15	3:20	3:25	3:30
Main Steam Temperature	915.00	950.00	940.00	920.00	1000.00	1050.00	1015.00
Reheat Steam Temperature	990.00	940.00	1000.00	1000.00	910.00	1010.00	1010.00
Gas Exit Temperature	669.00	675.00	668.00	665.00	665.00	662.00	662.00
Main Steam Pressure	1797.00	1797.00	1797.00	1797.00	1797.00	1797.00	1797.00
Excess O2	2.30	2.30	2.30	2.30	2.30	2.30	2.30
Steam Flow	135.00	136.00	136.00	136.00	136.00	136.00	135.00
Burner Tilt	35.00	30.00	30.00	30.00	30.00	30.00	30.00
Main Steam Attemporating Spray	0.00	0.00	0.00	0.00	0.00	0.00	0.00
Reheat Steam Attemporating Spray	0.00	0.00	0.00	0.00	0.00	0.00	0.00

VITA

Candidate's full name	:	Karen Stephens
Place and date of birth	:	Fredericton, N.B., Canada August 27, 1961
Permanent address	:	R.R. #1 Mouth of Vital York County, N.B. E0H 1N0
Schools attended	:	Fredericton High School Fredericton, N.B., Canada 1976-1979
Universities attended	:	University of New Brunswick Bachelor of Science Computer Science 1979-1983 Fredericton, N.B., Canada

Publications

Stephens, K. and Nickerson, B., "Temporal Reasoning Considerations for a Sootblowing Advisor", Proceedings of the 2nd UNB Applications of AI Workshop, Fredericton, N.B., Oct 3, 1989, pp.18-28.