RULE-BASED INTEGRATED BUILDING MANAGEMENT SYSTEMS

A thesis submitted for the degree of philosophiæ doctor

by

Gary George Clark

Department of *Electrical and Electronic Engineering, Brunel University*

JULY 1993

Give Instruction to a Wise Man ...

... and He will be Yet Wiser.

Proverbs 9.9.

ABSTRACT

The introduction of building management systems in large buildings have improved the control of building services and provided energy savings. However, current building management systems are limited by the physical level of integration of the building's services and the lack of intelligence provided in the control algorithms. This thesis proposes a new approach to the design and operation of building management systems using rule-based artificial intelligence techniques. The main aim of is to manage the services in the building in a more co-ordinated and intelligent manner than is possible by conventional techniques. This approach also aims to reduce the operational cost of the building by automatically tuning the energy consumption in accordance with occupancy profile of the building.

A rule-based design methodology is proposed for building management systems. The design adopts the integrated structure made possible by the introduction of a common communications network for building services. The 'intelligence' is coded in the form of rules in such a way that it is both independent of any specific building description and easy to facilitate subsequent modification and addition. This is achieved using an object-oriented approach and classifying the range of data available into defined classes. The rules are divided into two knowledge-bases which are concerned with the building's control and its facilities management respectively. A wide range of rule-based features are proposed to operate on this data structure and are classified in terms of the data classes on which they operate.

The concepts presented in this thesis were evaluated using software simulations, mathematical analysis and some hardware implementation. The conclusions of this work are that a rule-based building management system could provide significant enhancements over existing systems in terms of energy savings and improvements for both the building's management staff and its occupants.

DECLARATION

This thesis is the result of my own work and, except where explicitly stated in the text, includes nothing which is the outcome of work done in collaboration. No part of this thesis has been or is currently being submitted for a degree, diploma, or any other qualification at any other university.

ACKNOWLEDGEMENTS

The author is grateful to Dr Pratap Mehta for supervising this work and to Mr Tom Thomson for many useful discussions. The work formed part of a collaborative DTI/SERC Link initiative project between Brunel University, the insurance corporation Lloyd's of London, consulting engineers Dunwoody and Partners and BMS manufacturers Thorn Security. The author is particularly grateful to Lloyd's for providing data from their building to be used in the case studies and to Brunel's Estates Department for further information and access to their Trend control system.

List of Figures

Page

List of Tables **Page**

CHAPTER ONE

INTRODUCTION

1.0 Overview

The function of a building is to create an artificial environment to protect the activities taking place within it from outside influences. To meet the complex needs of modern work places computer-based technology is increasingly coming into the design, construction methods, materials and operation of buildings. The description 'intelligent building' has emerged to describe some buildings which incorporate these features. This thesis is concerned with the building management system (BMS), which is one component of an 'intelligent building'. The BMS has provided many enhancements in the control and management of buildings. However, the use of BMS has also resulted in some problems for the occupants and the staff whose function is to manage the building. In an attempt to overcome these problems and improve on current systems, this thesis proposes a design methodology for an intelligent integrated building management system. This chapter defines the function of a modern BMS within the wider field of 'intelligent buildings' and introduces the research work undertaken by the author.

1.1 Intelligent Buildings

The term 'intelligent building' has been in use since the early 1980s. The first intelligent building is claimed to have been built in America and was completed in 1983. Much of the drive behind intelligent buildings in the USA came not from the building operators, but from the equipment manufacturers, following the deregulation of the US telecommunication industry. Today the USA claims to have many intelligent buildings but the Japanese are world leaders in this field [1,2].

There has been considerable interest world-wide in the concept of the intelligent building but there is still no generally accepted definition of an intelligent building. The definition often depends on whom one is talking to. For example, an architect would consider a high-tech building to be one which is constructed in an innovative way. It might use advanced materials such as silicone glazing or incorporate sophisticated construction methods, such as pre-fabricated modules. Mechanical and electrical engineers consider a building with an automated building control system as being in the high-tech category, whereas the average building user considers that a building incorporating IT systems an intelligent building. None of these examples demonstrate that the building is itself 'intelligent', but serve to show how this expression has been widely misused.

One of the most quoted definitions for an intelligent building is that given by the Intelligent Building Institution in Washington:

> 'An intelligent or high tech building is one in which the building management system, the communication system and the information management system are integral to the building, and where these systems effectively manage resources in a co-ordinated manner to enhance the well-being and productivity of the occupants, and to enhance operating cost savings and flexibility.'

This definition of an intelligent building highlights the operating costs as being of paramount importance. A survey in the USA [3] indicated that an intelligent building costs somewhere between 8% and 10% more to construct than traditional buildings. If this is the case, for the concept to be viable, it must be attractive in revenue rather than capital terms. However, the operating cost is seldom considered early enough in the design programme. This can result in a building that not only costs more to commission, but also has a significantly higher operating cost to maintain. Tenants are often paying increased or even over-the-odds premiums for high-tech space. The only justification for occupying such space appears to be that it either suits or boosts the corporate image of the building's tenants.

To be worthy of the title 'intelligent', a building should be shown to make the best use of its structural envelope before resorting to energy consuming systems and high-tech equipment. The envelope of an ideal building needs to be flexible and adaptive. This is required to maintain a relatively constant building environment in circumstances of changing external conditions and varying activities of the occupants. Research has been conducted into the 'intelligent wall', in which the building's envelope automatically recognises and adapts to differing climatic conditions [4]. This is a very interesting idea and initial results show that it will shortly become technically feasible if not financially viable. The same is true for a wide range of new and emerging technologies appropriate for the cladding of the roof and walls. The ideal cladding would change its optical and insulating properties with changes in outdoor temperature and light.

If the 'intelligent wall' is not yet cost effective more use should be made of natural heating, ventilation and cooling. The new county hall for Conseil General des Boüché in the city of Marseilles is a good example of a building which does make the best use of its structural envelope. The building has been planned with external corridors. These act as buffer zones, moderating the effect of the external climate. In summer they prevent excessive solar glare and are naturally ventilated. In winter they act as an additional insulating layer. The capacity for heat storage by the building offers a daily strategy for reducing the peaks in the cooling load and discharging heat later. Daylight plays an important role not only in reducing the electrical lighting loads but also in improving the quality of the office environment. Fresh air is supplied from the roof level. This ventilates the building in the summer using cool air during the night time and is pre-heated via a roof top solar collector during winter. When compared to a building equipped with a variable air volume (VAV) system, the system running costs were substantially lower. Analysis by the building's structural and environmental engineers, Ove Arup and Partners, predicts lighting savings of 50%, refrigeration 66% and heating 50%. These may be slightly optimistic but they are very encouraging [5,6].

The number of structurally intelligent buildings, compared with the number of high tech buildings, is still very low. It must be assumed that in times of recession property developers are reluctant to experiment with new ideas and high-tech architectural techniques owing to the risk of failure. A building with an inefficient ventilation system can be cured by the commissioning of a new system but the structural design can rarely be changed without demolition.

1.2 Building Management Systems

In the vast majority of intelligent buildings the focus has been towards high technology control and management systems. These are usually centred around what is called the building management system. Building management systems (BMS) have developed from the simple applications of the 1970s to complicated integrated systems using complex software for total management of the environmental conditions in a building. In today's buildings the BMS can cover the monitoring and control of the majority of the building's services. Depending on the level of integration and sophistication within the building, the BMS can encompass any or all of the following areas [7,8,9,10].

- Heating, ventilation and air-conditioning control
- Lighting control systems
- Global energy management
- Load shedding and control of emergency power
- Fire management systems, including detection and smoke control devices[1]

---

[1] The above list includes fire management systems but it is uncommon for an automated fire protection system to be packages as part of the BMS unless special permission is obtained under local fire safety regulations.

- Security management and access control systems
- Time and attendance recording systems
- Miscellaneous building systems including pumps, sumps, etc.
- Elevator and escalator control systems
- PBX or telecommunications systems
- Error reporting and diagnostics
- Data archiving

A computerised maintenance system may also be included as an integral part of BMS operations, but it is more commonly provided as a separate feature to the BMS [11,12]. A BMS can provide considerable energy and cost savings compared with more simple equipment but it represents a considerable investment by the building owner [13].

The introduction of modem BMS can lead to significant improvements in the operational performance, energy efficiency and reliability of building services. Some claim that savings of more than 30% can be achieved by installing a BMS in an otherwise well serviced building [14]. With a national energy bill in 1990 of £4.8B[2] for environmental services in commercial and public sector buildings, such potential savings resulted in a buoyant market for BMS until the start of the recession[3]. However, one can question the criteria used to judge the performance of building energy management systems. If the energy consumption before the BMS was installed was twice what is should have been, due to inefficient equipment or incorrect settings, then claiming a saving of 30% is not really a major achievement: it is little more than the identification of obvious waste. It should also be noted that many energy management strategies are often expensive in terms of the life of the mechanical plant serving the environmental services. Where this is not recognised, and the energy management system is not integrated into the maintenance program, the direct impact each has on the other can mean that money which is saved in one area is lost in another [15].

Customers buy a BMS on the basis that it will make their building easier to run and provide many energy, and hence cost, saving features. The manufacturers like to give their customers the impression that high technology leads to the efficient running of buildings, with happy occupants and contented facilities managers with nothing to do but sit at their desks and watch their computer screens. In reality this is not the case. The use of high technology can lead to many more opportunities for things to go wrong and more complicated faults. This leaves the facilities managers with problems which they were never trained to solve [16].

---

[2] Source : The department of energy report 1990 HMSO.

[3] In a report published by PROPLAN in 1990, the UK IBMS market was reported to be worth £51.15 million in 1989 with a forecast of £117.00 million for 1994.

This thesis proposes a methodology whereby improvements can be made to the control and management functions within building management systems through integration and the use of artificial intelligence techniques. Such improvements would firstly lead to increased energy savings by tuning the energy consumption to the use of the building whilst causing least inconvenience to the occupants. Secondly, they would make the building easier to maintain by incorporating 'expert' knowledge into the management systems. The aim is to build some 'intelligence' into building management systems.

1.3 Background to this Work

In November 1990 Brunel University was successful in being awarded a DTI/SERC Link project in association with the insurance corporation Lloyd's of London, Dunwoody & Partners consulting Engineers and Thorn Security Ltd. The project's main aim was to investigate the use of rule-based KBS for integrating building services in high technology buildings. Brunel was the science partner and employed three research assistants to work on the project, one of whom was the author. Thorn Security was the manufacturing partner having recently bought up JEL, which made them a serious company in the field of building energy management systems. Lloyd's had experienced problems with the building control systems in their prestigious new high technology building constructed in 1986 (The Lloyd's 1986 building, referred to in future simply as the Lloyd's building). Lloyd's was thus a source of valuable information along with being an ideal test bed in which to try out some of the ideas proposed during the project. The author's involvement enabled him to gain valuable experience of some of the problems involved in high-technology buildings from the point of view of a building's owner, the building's occupants, the consulting engineers who specify building services, control room operators and a building control system manufacturer. It also provided the author with access to a great deal of information from the control systems in the Lloyd's building, some of which has been used in this research work.

1.4 Summary of Chapters

Chapter two explores some of the problems experienced with conventional building management systems. In recent years, in an attempt to overcome these problems, there has been a move towards the integration of building services. The concept of the integrated building management system (IBMS) is explored and in particular the arrival of a standard building local area network and smart sensors. Finally the chapter proposes that such a structure opens up many opportunities, which were previously impractical, for advanced control and management facilities using knowledge-based systems. What is not proposed is that the whole system should be replaced with one using 'all singing and all dancing' artificial intelligence (AI) controls and management. The author believes that there are advantages to be gained by adding AI only in specific areas and thus provide enhancements to the building management system without incurring excessive extra cost.

Chapter three identifies three distinct areas in which artificial intelligence can be applied to integrated building management systems: facilities management, control applications and assisted operator procedures. Each of these functions require information from the building and its systems. This knowledge can be fixed, dynamic or historical. In each case the rules which process the information must be independent of the actual data but instead operate on clearly defined data types. To facilitate this a general data structure is proposed. This is illustrated using data from the systems in the Lloyd's building in central London. The building is described in terms of objects. The information is broken down into five main classes and further divided into sub-classes. Objects inherit properties from class and sub-class levels. Using this general information structure, object-oriented rules can then be written for any of the three knowledge-based functions described previously. This is illustrated in the following two chapters using a series of case studies.

Chapter four describes how the data structure can be used for control applications. An integrated zone energy control module is proposed. This can operate autonomously from the IBMS and using a rule-based control strategy automatically tune itself to the zone under control. The chapter describes how such modules, when integrated within the IBMS structure, can be used for more advanced integrated energy control strategies. The current environmental status of each zone[4] (i.e. temperature, humidity, etc.), along with occupancy prediction, is fed into the global[5] 'building' database. This knowledge is used for applications such as global energy management, load shedding and load cycling control. In each case the aim is to minimise inconvenience to the occupants of the building whilst making cost savings. The ideas are explored using studies incorporating real systems data supplied by the estates department of the Lloyd's building.

Chapter five explores knowledge-based enhancements in the field of facilities management. Following a study of current building management systems an improved layout for the BMS operator's screens is proposed. This includes the introduction of three windows giving the operator status, faults and advice. The chapter concentrates on how the advice data can be provided using knowledge-based diagnosis and inference mechanisms. Two areas are

---

[4] In this thesis it is assumed that a building can be segmented into distinct areas for the purposes of local control; such an area is defined as a zone.

[5] In this thesis the term 'global' is defined as meaning the whole building or the involvement of all the zones within the building.

considered: diagnosing problems arising from BMS alarms and problems reported by people in the building. The aim is to build more intelligence into the BMS monitoring systems and hence to reduce the level of expertise required by the operator. This is achieved using a rule-based diagnosis system. The diagnosis rules are object-oriented to enable rules for different classes of equipment to be added or updated without affecting the overall inference mechanism. The fault diagnosis work is extended to fault prediction. Finally, a method is proposed for using the maintenance and control information, both current and historical, for the optimum tuning of preventative maintenance scheduling.

The final chapter summarises the work undertaken and emphasises the areas where a rule-based IBMS can reduce the building's operating and maintenance costs. The possible problems and limitations of the approach are discussed and areas for further work in this field are proposed.

CHAPTER TWO

THE INTEGRATED BMS - AN EMERGING CONCEPT

2.0 Overview

The concept of the intelligent building has been steadily evolving for some years. From the early days of the first applications of 'intelligent' computer technology to buildings, the emphasis has progressively moved from automating individual building services, towards integrating all the building's systems, via centralised building management systems (BMS). This will be achieved through the use of an integrated building data-bus. This chapter summarises the emerging concept of the IBMS and discusses the possibilities of providing it with artificial intelligence through the use of knowledge-based system (KBS) technology.

2.1 The Move Towards Integration

The need for a move towards the integration of building systems is based on the fact that it is common for consultants to specify building-services equipment from many different suppliers. Therefore, there exists a communications problem between all the different systems. One reason for this is that it is very uncommon for any one supplier to produce the best fire, security and energy management systems all within the same product range. Another reason may be that building owners lack confidence in any one supplier to continue to offer the same level of performance on a long-term basis. A good example of this is Lloyd's BMS which includes many high-tech features but lacking in integration. The systems in this building are supplied by many suppliers using different types of computer hardware, each running different operating systems[6], with no provision for local-area networking or a general sharing of data. As a result, each system provides autonomous control with little integration with any other system, except in the event of fire when the HVAC (heating, ventilation and air conditioning) systems will stop the escalators and lifts. This lack of integration is particularly noticeable in the BMS control room where the operators sit in front of an array of control terminals each having different passwords, menus and graphics.

---

[6] The main BMS is run on a Texas Instrument computer; the lighting-control system uses a Digital Equipment PDP11; the lifts and transport controller uses a dedicated processor-board based around the Intel family of computers; the maintenance management system is run on a 80386 based personal computer, running a multitasking operating system. None of these systems communicates with any other.

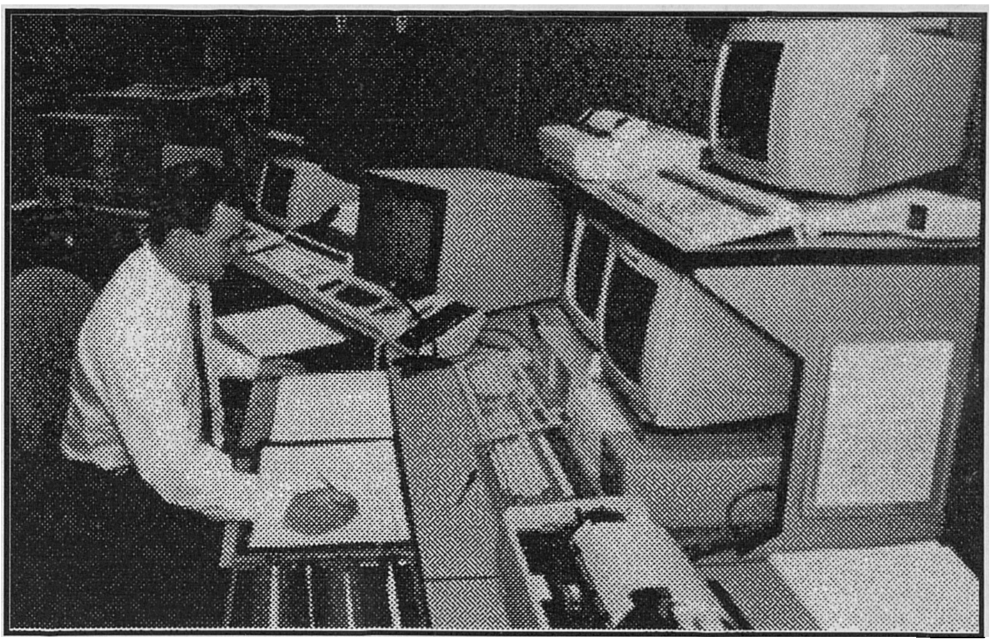


Fig 2.1 Photo of Lloyd's Building BMS Control room
(Reproduced by kind permission of Lloyd's of London, photographer: Colin Jones)

In the Lloyd's building the control room, shown in Fig.2.1, is manned by two operators for 24 hours a day. They sit at a desk much akin to the control room in a power station surrounded by seven VDU screens. There are separate VDUs for the Johnson's Controls system, the ECS energy management system, the Express lifts control and the CALMS maintenance management system. Besides all these there is a MIMIC display for the electrical systems and numerous printers.

Some companies have tried to overcome this central communications problem through the use of gateways [17]. Gateways aim to achieve integration at the supervisory-computer level of the systems by attempting to make one computer manage and display the data collected from several systems. The Windows 3 PC operating-system environment from Microsoft is particularly suited to this task, allowing many gateways to be run simultaneously on one computer and to be displayed on one monitor. Gateways are effectively software interfaces providing protocol translation at a fairly high level, but even with them it has not proved easy to 'mix and match' systems designed by different manufacturers. Gateways must be seen as a compromise solution: the most efficient solution would be for all the building systems to operate on a single local-area-network (LAN), communicating via a defined standard protocol.

There are two important aspects to note. Firstly, an individual service such as lighting, must perform to set requirements and secondly, various services must interrelate. For example, if there is a fire in the building all the other services must react in a predetermined manner so that the building is evacuated quickly and the fire controlled. This can be accomplished if the services are controlled by an integrated building management system (IBMS). This can be done in such a way that appropriate information and advice can be provided for action by the various levels of staff running the building· Existing systems provide various levels of integration but the overall integration still remains an emergent concept, particularly in the UK.

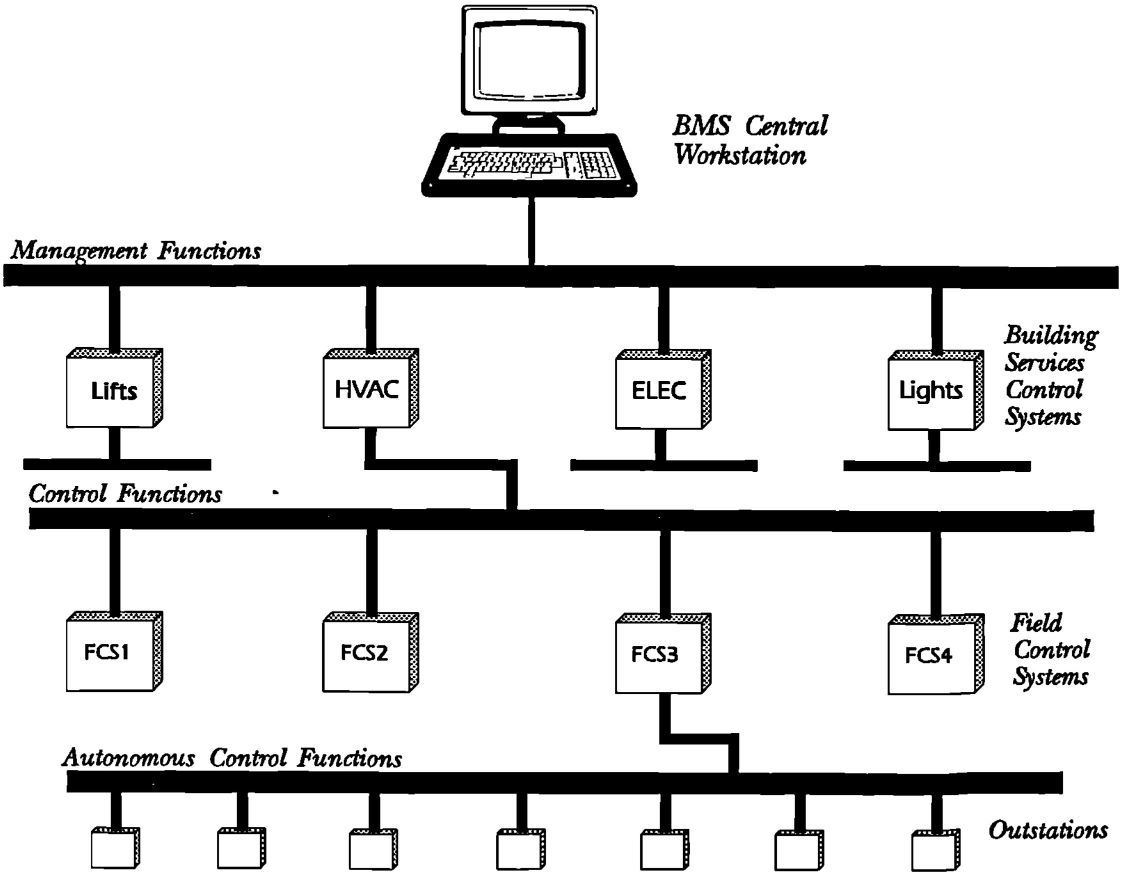


Fig 2.2 Integrated Building Management System
(previously published in ref.69, reproduced here by the kind permission of the co-authors)

The early designs of integrated building management systems merely collected the outputs from the local controllers dedicated to fire protection, security, HVAC and energy management for central monitoring of all these services [Fig.2.2]. Basically, these inputs to the central computer were exceptional inputs, i.e. showing deviations from the norm which required the operator's attention. Thus integration was an overlay on the existing systems, requiring substantial additional wiring and offering improved management response and automatic logging of system actions. In the case of a computer or communications failure the essential local control functions were still operative.

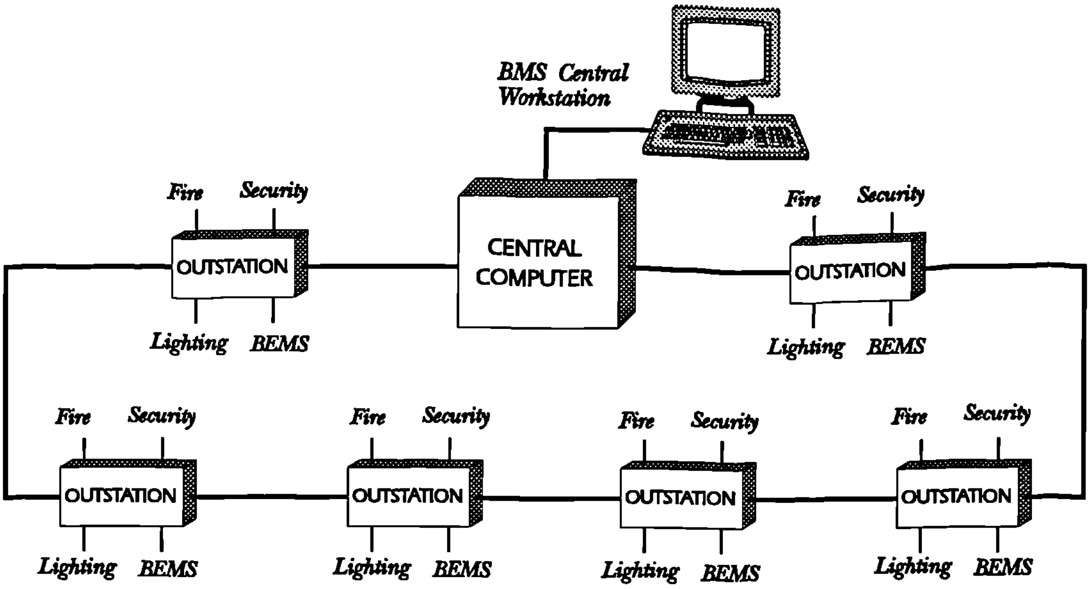


Fig.2.3 Integration with Distributed Intelligence

A more advanced approach is to integrate the systems using distributed intelligence. This is emerging through the use of microprocessors in local controllers, enabling many more inputs to be monitored and more functions to be performed for less cost. Such controllers are known as outstations and are connected to the central IBMS computer as shown in Fig.2.3. This architecture provides the central control computer with the information required for more advanced global control and monitoring functions. The centralised services carry out a series of major global control functions, monitor all events, notify the operator of any abnormal conditions and provide advice. It can also support the building's management activity by providing a preventative maintenance schedule.

2.1.1 The Building Data-Bus

Integration of building services should reduce costs in two ways: first by unifying the various wiring to the control systems (possibly with separate but interconnected circuits for fire alarms); secondly by standardising components such as switches, controllers and sensors. Currently, buildings incorporate at least four layers of wiring: voice, security data, environmental control and word processing. To reduce this in the future, the move will be towards the building data-bus. A common building data-bus should bring down the cost of controlling building services and make complex building management systems more widely available. The current proposal for a building data-bus is that it should be essentially separate from other data networks in the building. In the future it could include other data networks such as computer and telephone communication links. However, this may cause problems in shared tenant buildings where the various occupiers may want their own separate networks for security reasons.

Standardisation should mean that components are manufactured in longer runs and are interchangeable, so that no manufacturer can put a premium price on a product because it is the only one that can be used in a particular system. Individual switches and sensors will have to be designed to incorporate communication circuitry, so that they can send and receive signals which are compatible with all other equipment connected to the network.

If physical integration, supported by standard protocols or by protocol conversion as appropriate, exists between systems a truly integrated approach to building management can be applied. The introduction of the building data-bus may not in itself increase the sophistication and accuracy of control achieved in buildings, but should make control systems cheaper and easier to install and operate. This could provide advantages in the control and operation of all of a building's services. Lighting levels could be accurately controlled and environmental conditions precisely adjusted to meet the requirements of the occupants of the building. Such an IBMS would be able to react quickly to changing conditions, make optimum use of energy, and control maximum demand for electricity. For example, given a requirement to reduce electrical usage under either a peak-demand or time-of-day billing regime, the potential overlay of an electric load-control scheme on an IBMS could accomplish the required load reduction without any significant disruption to services or discomfort to the occupants. In this way a fraction of the lighting load could be reduced by selective reduction of lighting levels at the perimeter through either dimming or split wattage fixtures, assuming sufficient daylight is available. Lift speeds could be reduced, or a few lifts temporarily parked. The HVAC load could be temporarily diminished by altering control-loop set points incrementally rather than through a wholesale shutdown of equipment, while generators or uninterruptible power supply (UPS) equipment could be brought into service. Occupancy schedules, either programmed or monitored by the building's access-control system, would automatically alter the load reduction strategy as appropriate.

An outstanding problem is how fire-alarm circuits should be dealt with. Theoretically, fire-alarm sensors and associated equipment could be connected to the data-bus together with all the other plant. However, current fire regulations and the requirements of local fire officers mean that dedicated circuits have to be provided for the fire system[7]. Exactly how the problem should be resolved is currently the subject of discussion but it seems likely that, at least in the first data-bus installations, there will be separate circuits interconnected with the main system for smoke control and similar functions.

---

[7] British Standard BS5839, 1988

The other remaining problem is exactly which data-bus standard to use. In Europe, a race for a common standard for building control systems is currently under way between two rival Consortia, promoting the European Installation Bus (EIB) and Batibus standards respectively. Siemens, ABB, the Pillar Group (including Trend, Gent and MK Electric), Crabtree and Scholes are some of the members of the Pro-EIB Association, while Merlin Gerin, Landis & Gyr, Telemecanique, Philips Lighting and the Pillar Group (again) are in the Batibus club of companies (Landis and Gyr are also co-operating in another initiative called Profibus). Both of these systems are relatively low technology being switched pair wire. A further proposed standard called factory neutral data (FND) transmission is becoming popular and is sponsored by the German government.

Batibus can be configured in any combination of star or tree loop structures with maximum lengths of 2500m for 2.5mm^2 cable. It can support 1000 points, up to 75 of which can be powered by the bus. The network can accommodate 240 addresses, where an address represents a single device or group of devices. Up to 20 messages a second can be transmitted, at an average transmission time of 0.2 seconds. The Eibus uses a tree like topology and for larger installations 12 branch busses can be connected to a thirteenth backbone bus. Each branch can carry up to 64 devices, giving a theoretical capacity of nearly 8000 devices. Devices connected to the bus have to incorporate a chip to allow them to send and receive signals. Merlin Gerin has Intel and NEC manufacturing Batibus chips; Siemens are manufacturing the EIbus chips. Both bus systems use carrier sense multiple access (CSMA) protocol, which causes required devices to first 'listen' to the line, to ensure it is free, before sending. The third system FND allows intermediate network and interfacing of equipment. This enables limited operation from one BMS of data points connected to a BMS of another manufacturer also supporting the FND specification.

The current form of the data-bus may not be the final solution. In data technology terms twisted pair technology is old fashioned. Some modern buildings are being equipped with broad band communications networks, based on coaxial or fibre optic cable. Hence at least two data systems could be installed in the modern building of the future, one based on twisted pair technology and the other probably fibre optics. It is hoped that a standard will soon be laid down by the European harmonisation organisation CENELEC or in America by the ASHRAE SPC 135P standards project committee [18,19]. If the delay continues, the eventual standard may be market led, resulting in a slower process, as in the case with satellite television and the VHS/Betamax video standards.

2.1.2 Intelligent Sensors

Associated with the idea of IBMS and the standard building data-bus is the concept of intelligent sensors. These provide distributed intelligence and the ability to transmit information to the central building automation system. The basic idea of an intelligent sensor is one constructed from a single piece of silicon, capable of intelligent analysis and processing of the physical quantity being sensed. The intelligent sensor approach has the potential to provide distinct advantages over centralised signal processing: the sensed value can be converted and transmitted to other devices as a digital signal with improved accuracy and long term performance. By providing signal processing on an individual sensing element basis, data unique to the building or even the sensor's location and detection mode, can be stored with the sensor at the time of production. For example, the sensor could incorporate calibration curves. Eventually, these sensors can be very low cost owing to silicon mass-production techniques, but the initial development costs and overhead could be significant [20]. If the development costs for the single chip idea are too high in the short term, the sensor can be implemented using several separate mass produced chips along with a sensing element. Control systems incorporating intelligent sensors are beginning to appear and the introduction of a standard building data-bus should make them more widely accessible [21].

Future research lies in the intelligent processing of all the information available from the building's sensors. There are some systems available today with a relatively high degree of intelligence. For example, modern analogue addressable fire sensors are designed to analyse the amount of smoke present. A further extension would be a rough chemical analysis of all airborne particles which would enable the correct action to be taken against different types of fire. Also, analysis of data from several different sensors reduces the occurrence of false-alarms. This is currently a big problem for manufacturers of fire and security equipment[8] as the majority of alarms are false alarms and cost a great deal of money to service.

2.2 Knowledge-Based IBMS

With increased integration, networking and intelligent sensors the facilities managers of large buildings now have more data supplied to them than ever before. However, there is a real danger of them being 'drowned in data' while at the same time starved of useful information. Facilities management is a widely ranging skilled job based around managing the building's

---

[8] This was reported as a serious problem by the Director Corporate of Thorn Security. The police forces, who have to cope with false alarms, have a significant influence on the specifications and quality of intruder installations. They have approved lists of installers who they recommend and blacklists of installations which are causing trouble through repeated false alarms.

operational structure and its services. As the move towards high-tech buildings increases, the facilities manger's job is also encompassing managing the building's information technology systems. In addition, the facilities manager must understand their organisation's business and be able to assess the effect of failing to provide the required environment. They must also understand the technology employed to control the building and be able to interpret and act on the information it provides. It is pointed out by representatives of the profession that the job is increasingly becoming more complicated due to these multi-disciplinary requirements [22]. It may be that not many facilities managers wholly fit these demanding criteria and in the future organisations using high tech buildings will have to look hard to find the appropriate personnel [23].

There is currently a great deal of interest in such artificial intelligence (AI) technology and in many areas it is moving out of the research laboratories into practical applications. Knowledge-based systems (KBS) are one form of AI where knowledge is captured from experts and stored as a knowledge-base. A KBS is capable of accessing the knowledge-base, diagnosing the problem and suggesting the best course of action in each case. An intelligent building control system, with a KBS reacting to information fed from the various control sensors in the building, is now a possibility.

A non-KBS IBMS can quite easily cope with the switching of chiller plant to meet increased cooling requirements. However, the average BMS is simply not intelligent enough to give a range of possible reasons for failure; this is where KBS can improve on existing systems. In the event of a breakdown or problem, the KBS can analyse the series of events leading up to the incident and diagnose the cause. Based on this diagnosis, it can suggest the best course of action to undertake. Such features do not appear in current building management systems.

Time was spent in the Lloyd's building BMS control room, observing the operators at work and their interaction with the computer systems. Contract maintenance staff from Johnson's and Express lifts are employed in the building 24 hours a day in addition to 60 of Lloyd's own maintenance engineers. The control room operators are in contact with the maintenance staff by two way radio. If there is a problem, either reported over the phone by an occupant or by the computer systems, the events which followed sometimes became somewhat chaotic. Whilst observing the operators a problem occurred with the elevator doors not opening on one particular tower. The trapped person was a senior underwriter on his way to an important meeting and this caused some panic in the control room. The operator dispatched maintenance staff to the tower whilst attempting to investigate the cause of the problem. The lift control system has a text based diagnostics screen. To use this system involved logging into the system with a set of passwords then looking through a list of numeric fault codes. The system is not very user friendly with fault codes having to be looked up in a reference manual. Whilst this was happening another call informed the operator that a chiller was refusing to start. The operator then logged into another system at the other end of the bench and worked his way through a long series of menus to get to the boiler control in the attempt to override the time control setting and start it manually. This having failed, the operator decided to send a Lloyd's maintenance engineer to investigate. Concurrently another call informed the operator that the underwriters in gallery four were complaining of a lack of heating. The operator resorted to a mass of computer printouts to look up the air handling unit associated with that particular area in order to send an engineer there. The engineer looking at the boiler reported back that the problems were not with the boiler itself, but with the BMS control system, as switching to manual override started it. He was not qualified to look at the controls and a control engineer was dispatched. This meant that half an hour was wasted in sending the wrong class of maintenance engineer. The overall impression was of a somewhat chaotic time for the operators and that the computer systems could do a lot more to assist in their work. Some thoughts came to mind:

- The systems should be organised such that the operator was only looking at one screen. It is virtually impossible to monitor seven screens simultaneously.
- The monitor screen should display only the information required whilst all the current monitoring data should be available if requested.
- The diagnosis information should be in plain, easy to understand English.
- The operator should not have to resort to computer listings or look up tables to diagnose a problem, all relevant data should be available on the screen.
- The systems should provide advice to the operator based on the reported symptoms. This could include suggestions on the cause of the problem, a list of possible actions to take and which class of maintenance engineer to send.

If carefully programmed, by capturing and using knowledge of experienced facilities managers and mechanical or electrical engineers, an intelligent building control system can safeguard a high-technology building and its components 24 hours a day: diagnosing faults, drawing on reserve equipment and alerting maintenance teams as necessary. This would reduce the need to have highly experienced facilities managers and engineers in the building full time. Lloyd's have two full time engineering staff monitoring the building management system for 24 hours a day. It is very wasteful to tie up experienced and highly paid personnel for a job that is very mundane for the majority of the time. They would be better deployed in optimising the operation of the building services rather than monitoring them.

The primary objective of an IBMS is the optimisation of control to balance the requirements for occupant's comfort, energy efficiency and reliability. Systems found in today's buildings do not fully justify the label 'intelligent' because they do not fully possess those features that should be associated with 'intelligent' control: namely an appropriate self-learning facility that can provide optimum control and an ability for individual control loops to interact, so as to improve the performance of the system overall. Research has shown that improvements can be made by combining some of the more conventional control techniques, such as proportional-integral-derivative (PID) control, with AI techniques [24,25]. Currently PID controls comprise the majority of control loops associated with building services. In practice, the control parameters are ether set to default values, in which case optimum settings are unlikely to be achieved, or are manually tuned, which requires the attention of an experienced engineer. Even with this attention, changes in the characteristics of the system being controlled can mean that subsequent operation becomes sub-optimal.

The application of AI technology in conjunction with modem conventional control techniques to produce self-tuning controllers has the potential to reduce the time and hence cost required to manually commission and subsequently re-adjust control loops. It can enable self-tuning operation (automatic commissioning and subsequent self-adaptation), as well as the continual revision of control parameters beyond the commissioning phase. This thus gives better long-term control performance through the automatic selection of 'optimal' control parameters and so provides more consistent control, improved occupant comfort, and reduced energy use. In addition, a KBS approach can provide greater diagnostic feedback to users on the state of the system.

An IBMS which includes maintenance management can make use of the automatic transfer of monitored information. This data is collected from the building control systems at the sensor level and passed to the maintenance management system on a regular or real time basis. Information can be collected on plant performance in terms of running time, efficiency, flow rates and power consumption. Other parameters and trends may be measured or calculated. This data can be transmitted through the local area network to the BMS historical database to assist in the operation of planned preventive maintenance. Faults, breakdowns and alarms would also be automatically signalled to the IBMS. This avoids the need for a manual input to the database, thus ensuring that it always has the latest and the most accurate data. Should an alarm occur which requires attention, the appropriate message and work order can be printed out automatically. The urgency of the alarm can be categorised so that the time scale over which corrective action is taken is appropriate to the situation. Historical data can also be called upon to assist the maintenance engineers to plan future strategies.

The buildings use and the requirements of its occupants will inevitable change with time. Hence the maintenance work and the priorities allocated to such work should also change with time. If the maintenance management system is also integrated into a KBS type IBMS, then the planned maintenance period could become a dynamic variable, adjusted to achieve the most economic schedule. For example, if a chiller system breaks down on average once a month and its planned preventative maintenance takes place every two years, there is a clear case for carrying out preventative maintenance more frequently. In most buildings the maintenance interval for equipment is fixed and can only be changed by direct input from the user. In the case of the Lloyd's building, where more than thirty-five thousand maintenance jobs were undertaken in 1990 alone, the task of identifying every area of intelligent feedback into the maintenance program by manual means would not be cost effective. An intelligent IBMS could fine tune itself to best manage the building and adapt as the system requirements change throughout its life. By using a KBS approach to this 'intelligent tuning', the system can explain why maintenance is scheduled at a particular time by use of engineering rules, actual breakdown data and statistical analysis. This would be of particular use to the maintenance and engineering staff within the building who may not have faith in KBS 'judgements' without knowing the facts behind its decision making. KBS in particular lend themselves to explaining the reasons behind decisions.

There are several opportunities for reducing costs and improving services in the area of emergency repairs or unanticipated failures. One opportunity lies in the ability to predict failure and correct it before it occurs [26]. This is a capability made viable by the integrated control concepts and condition monitoring possibilities which are becoming available in intelligent buildings. Effective preventative maintenance can reduce the number of failures, but it must be carefully considered because it can be more costly than failure driven maintenance. Faced with finite maintenance resources, the KBS type IBMS can optimise maintenance scheduling on the basis of priorities. For example, if two separate pieces of maintenance work should be undertaken during the same period, but in fact there are insufficient resources to undertake both of them, the KBS can re-schedule one of them based on economic or other factors and then explain its reasoning. The KBS can also be used for predicting breakdowns, maintenance cost and spare part requirements. This is particularly useful for planning in terms of budget requests and ordering materials. The system can advise on re-order times and the necessary stock of spares based on predicted requirements. It can also advise on specific maintenance requirements following a breakdown, highlight common or particularly costly faults, as well as providing 'decision support' criteria for repair or replacement.

2.3 Other Research

It has been recognised that the administration and control of maintenance is an increasing problem for the facilities managers of large buildings [27,28,29,30,31,32]. Most commercial maintenance packages consist of an asset register and a scheduling program. Few seem to integrate rule-based diagnosis within the maintenance management system. Possibly one of the most advanced maintenance management software packages is EASE (Event Actuated Support of Equipment) by Cormac Systems [33] which claims to use knowledge based diagnosis. However, EASE has some limited rule-based features which automatically classify problems into engineering disciplines such as HVAC, electrical or mechanical providing the user selects the problem description from a pre-defined list of options. No on-line diagnosis is available for specific types of equipment based on symptoms and past maintenance history. Cormac Systems have recently introduced an analysis package as an add on to EASE called SEQUAL (System for Efficiency, Quality, Utilisation and Availability Logging). This provides extensive reporting facilities for the management in areas such as availability, efficiency and utilisation of equipment. Both systems are limited by the fact that all data must be entered manually into the systems by what Cormac calls DCTs (shop floor data capture terminals). Operators around the plant have to enter operational events and incidents such as "machine idle", "waiting for fitter", "line changeover" as and when they happen. This introduces a laborious human element into the systems. A more advanced approach would be to integrate the maintenance management systems as part of the IBMS and all this data would be automatically on-line.

Research has been undertaken by Cornwall County Council where they monitored the temperature profile of a number of buildings using software developed by Stark Associates [34]. This provided data collection and analysis on the BMS output. The use of KBS for the interpretation and selective presentation of data from BMS has been shown to be a viable proposition, in particular the BREXBASII system developed by Shaw [35]. Here the Building research Establishment has installed a prototype system on-line for a year in Sir William Atkins House (SWAH) in Epsom. The findings suggest that a greater consideration of the central plant performance trend analysis might prove a profitable approach in future systems [36]. These would give the BMS the intelligence to carry out its own information and advice generation; something few BMS owners seem to have the time or the suitable skills to achieve in recent years [37].

Finley [38] discusses the possibility of integrating KBS into planned maintenance programs to interpret conditioning monitoring and provide technical support. Other authors specifically target the building services environment as an area where KBS can be usefully deployed

[39,40]. Diagnosis and maintenance for complex HVAC systems is a particular area of discussion as the current maintenance costs are specifically high and there have been some prototype systems developed in recent years [41,42,43,44]. Arueti [45] reports on a knowledge-based prototype for optimisation of preventative maintenance scheduling developed for a nuclear power plant. This is based on probabilistic judgement and inference rules, using data on failure rates, repair times, normal costs and other indirect economic costs such as accidental power loss. Similar approaches have been used for job scheduling within a manufacturing environment [46,47,48].

Fault diagnosis is a field where knowledge-based systems have been proven to be very useful. Applications cover a wide range of mechanical and electrical systems. Simple examples of this technology have been used in modern cars to monitor engine performance and diagnose problems. Many commercial electronic instruments now include a form of KBS diagnosis for maintenance. Faults can be diagnosed at system or component level, thus reducing the level of training required for the service engineer [49]. In some cases the aim is not to diagnose the fault precisely, but to reduce time in taking measurements and to provide an optimal inspection sequence [50]. There are many examples of applications in power systems for fault finding in large electrical networks. Some use high level languages [51] and some utilise KBS shells [52]. The use of object-oriented data representation for knowledge-based systems is becoming popular for many engineering problems. Here the data is stored with knowledge and structured around real world objects [53].

In some applications a need has been recognised to move away from the traditional 'shallow' rule-based fault diagnosis systems towards second generation 'deep knowledge' systems [54,55,56]. Instead of empirically finding associations between symptoms, faults and their causes represented in the form of rules a deep knowledge system attempts to diagnose faults based on a model of the system. A shallow system can suffer from limitations and inconsistencies in the rule-base when situations appear which were not considered at the knowledge acquisition time. A second generation system can overcome this limitation. Practical experience has shown that improvements can be made in diagnosis although using this approach often results in long response times. Research is underway into combining these two approaches into a hybrid system supporting model based diagnosis with explanation based learning and logical inferences [57]. A current area of activity is in integrating artificial neural networks and KBS in fault diagnosis [58,59]. Work has shown that a pattern recognition approach can be used to detect symptom patterns and correctly identify faults [60]. In some cases this has been shown to improve on both model based methods and KBS. Papers by Kurmar and Kamens both report on a successful application for fault detection and diagnosis in process automation [61,62]. This leads to a self-learning and adaptive diagnosis system, which is what would be required to improve on the skill of an experienced BMS control room operator.

Buildings have so far seen very little application of intelligent control systems. There has been work conducted in this area by the technical University of Delft where they have designed a system which predicts the local weather conditions and automatically opens or closes office windows while also adjusting the heating. As natural lighting becomes adequate artificial lighting is switched off. This system, which they hope to market, also controls blinds to modify shading [63].

Japan boasts the first intelligent house which has been build as a demonstration project in Roppongi, Tokyo in a small Nippon Homes housing development. This house has been developed as part of the TRON project - The Real Time Operating Nucleus led by a research team at Tokyo University [64]. The building is fitted with internal temperature, humidity, air speed and human presence sensors. These are linked with external weather sensors for optimised environmental control. Automatic windows provide fresh air ventilation in favourable weather conditions and under floor heating for cold conditions. The human presence sensors are linked to a security system which is programmed to recognise which parts of the house are occupied and when the house is empty it locks the main door, shuts all windows and activates the security alarm system. The house incorporates advanced lighting and sound systems. Combinations of light levels and background music can be programmes for the desired ambience. It even incorporates a computerised kitchen, with what is termed a cooking support system, utilising a KBS approach to help the cook choose and prepare seasonal dishes based on video demonstrations. The cooking appliances are linked to the computer which sets them automatically to the correct temperatures and cook time. Although technologically advanced, apart from the cooking support system, there is very little application of AI in terms of control and management of the house and its services.

KBS have been used for computer aided design in selecting building services at the commissioning stage. Fazio reports on work involving their use as an advisor in the preliminary stages of HVAC design [65,66]. This assists the designer to configure and size HVAC equipment for the exact requirements of the building. Factors taken into account include climatic conditions, the building type, the indoor/outdoor requirements and the level of indoor air pollution. Whilst work has been reported on energy efficient design and management, little work has been done into rule-based control of building management systems [67,68,69]

CHAPTER THREE

KNOWLEDGE-BASED APPROACH

3.0 Overview

To be practical, a rule-based building control and management system must be designed in such a way that the knowledge, coded as rules, is independent of any specific building. It must also be adaptive to changes in the configuration of the building or its services. The rules must operate on the data and information types within a general building description which are consistent irrespective of any one building description. This can be achieved with an object-oriented knowledge-based structure, where rules operate on objects within clearly defined classes. The values of the properties belonging to these objects are specific to the building and its control systems, but independent of the knowledge coded in the form of rules. This chapter proposes such an object-oriented data-base structure for integrated building management systems. Such a structure can be used for rule-based applications in the fields of both control and facilities management.

3.1 Proposed Approach

Object-oriented programming (OOP) is a method of programming that closely mimics the way things are done in real life. It is more structured than previous attempts at structured programming and more modular and abstract than previous attempts as data abstraction. An object-oriented database system stores both data and programs associated with an object. This is in contrast to programs that manipulate data in semantic data model systems, where the programs themselves are not part of the database system. In recent years object-oriented databases have been proposed for many engineering and knowledge-based applications [70,71,72,73]. The three main properties which characterise an object-oriented programming language are:

- Encapsulation: Combining a record with procedures and functions that use it to form a new data type called an object. It is possible to change the implementation of an object without altering the way it behaves. This makes the application programs independent of the representation of the objects they use.

- Inheritance: Defining an object and then using it to build a hierarchy of descendant objects, with each descendant inheriting access to all its ancestors' code and data.

- Polymorphism: Giving an action one name that is shared up and down an object hierarchy, with each object in the hierarchy implementing the action in a way appropriate to itself.

The main benefit of OOP is not in reduced development time; object-oriented development may take more time than conventional development as it is intended to promote future reuse and reduce downstream errors and maintenance. Changes to object-oriented development are easier and faster than with conventional development because revisions are more localised. It is maintenance and re-development time that accounts for as much as 60% of the development effort for conventional software.

Many object-oriented languages have been developed in recent years. One of the earliest was Smalltalk-80 which was developed in 1983. Today the language C++ is very popular. C++ is an extension of C proposed in 1986 and supports a range of data abstraction and object-oriented features without reducing too much the flexibility of C. It is now available on a large range of hardware and software platforms and is likely to be the most widely used object-oriented programming language for some time to come.

An object is the key item of information in the object-oriented representation. It represents any person, place, thing or idea in the domain of the particular application. One can describe an application's world in terms of various objects. For example, each particular area within a building can be defined as an object along with all the components needed to service the building. A class is merely a grouping or generalisation of a set of objects with some common properties. Objects are specific members of a class. For example, all the lights within a building may fall into the class of 'Lighting'. To avoid confusion in terminology, in this thesis a class name will be referred to in quotations marks and an object name will be indicated by the use of italics.

Objects may belong to several classes. A *fan* can be a member of the class 'Air handling units' as well as the more general class 'HVAC'. Classes may also have many objects and there is the possibility of many different relationships. A class can also have sub-classes. A sub-class is a class which represents a sub-set or 'specialisation' of another class. It is a class in its own right and has all the characteristics of other classes. For instance, 'HVAC equipment' could be one class with 'Pumps', 'Fans' and 'Boilers' as sub-classes.

Properties are used to describe the attributes of both objects and classes and one can use any number of properties in this description. For example, a *fan* may have a particular power rating and size. Both of these attributes are properties of the *fan*. Properties have a particular data type, most language implementations allow string, integer, float, Boolean, date, or time data types. While objects and classes may have specific properties, these are not limited to any one object or class. Thus other objects and classes can have the same property. Furthermore, since the property is independent of the object or class, it will always have the same data type throughout the knowledge-base. A property value representing the power rating of the object *lamp* can be represented as *lamp.power*. Properties are automatically inherited from a class to an object when the object becomes a member of that class. Thus, if a new object *lamp2* is created and defined as being a member of the class 'Lighting', all the properties belonging to the class 'Lighting' are inherited to the new object. The definition of this data structure is illustrated in Fig.3.1.

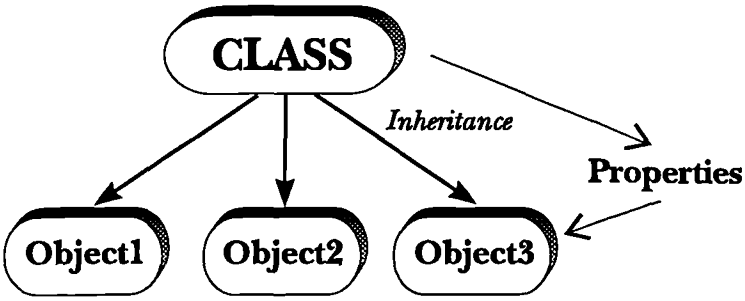


Fig.3.1 Knowledge Representation Structure

Each class definition includes properties which are inherited to the objects belonging to that class. There are many different ways in which a property can obtain its value. In this thesis, the properties of all the objects used fall into five categories defined by the author, namely: measured, set, status, actual and predicted. Every property's name used by the rule-base ends in a string which signifies how the value of the property is found. A description of the five property types is given here and Appendix A gives the a full listing for the defined classes, objects and properties.

(a) A measured property is one in which the value is obtained directly from a sensor. Such properties include light levels, power requirements, temperature and time. The sensors are linked to the building data bus and these properties are updated at regular intervals by polling the outstations. This is outlined later in chapter four.

(b) A set property is one which is defined before the rule-base is processed; it is not changed or modified through processing the rules. Modifications to set values can only be made externally to the rule-base either by the IBMS operator, or by associated (non rule-based), IBMS software. Many of the set properties are obtained from historical data stored in the IBMS central database.

(c) A status property is one that is defined following the result of a rule's evaluation. Until it is evaluated, a status property has the value 'Unknown'. It can take on any value or meaning when evaluated by a rule action. Finding the value of a status property may require backward chaining to other rules.

(d) An actual property is one which is calculated by means of a 'method'. A method is defined as an algorithm, or a function of the other properties, which is defined specifically for a class of objects. The method calculates the value of the property and the result is treated by the rules a being a true value or fact.

(e) A predicted property is similar to the actual property and is also calculated using a method. However, it is not assumed that the value is necessarily correct, thus predicted properties are treated with more caution when processed by the rules.

The implementation of these properties in the software simulation is outlined in Appendix B.

3.2 Object-oriented database

Using the object-oriented data representation as described previously, a data structure is proposed for a rule-based integrated building management system. Five main categories of data have been defined: 'Plant equipment', 'Building', 'Maintenance', 'Resources' and 'Planned events'. These are represented within the system as classes and are outlined below.

(a) The 'Plant equipment' class defines all of the building's services as objects. This class is divided into six sub-classes: 'Security', 'Transport', 'HVAC', 'Electrical supply', 'Lighting' and 'Fire prevention'. Properties describe the plant equipment and its control functions. There also exists measured properties, these store data on the current running status of plant such as the power consumption and historical plant data such as the last maintenance period and the total running hours since this maintenance.

(b) The 'Building' class stores a description of the building along with the current environmental conditions. Every distinct area within the building is defined as an object. This enables the rules to interpret the location data for the plant equipment along with other location specific data such as the temperature and occupancy of various areas. These properties are linked to dynamic measured data from the various room sensors.

(c) The 'Maintenance' data is divided into two sub-classes which store historical data on faults and breakdowns for the plant equipment in the building. Every fault and item of repair work is classified into a finite number of jobs. For each job the average time, frequency and cost are defined as set properties. Breakdown history is used to calculate the predicted breakdown probability for plant equipment using methods.

(d) The 'Resources' class defines the data objects which represent the resources available to the IBMS. It is classified into sub-classes of energy, maintenance staff and spare parts. This data enables the IBMS to predict what capacity there is available to undertake maintenance work and to optimise energy management in the event of supply failure. It also provides useful costing information for the building's management.

(e) The 'Planned events' database takes the form of a calendar for listing what work or events are scheduled for the building and its maintenance staff. Every planned event is defined as an object with properties describing the event. As planned events are specified by the IBMS operators their properties are mainly set types. Knowledge of planned events is used by the rules to modify the planned control of the building's services as well as the optimisation of planned preventative maintenance.

Almost all the specific classes of data outlined here are already available within a building containing individual control systems but in a distributed form. They are not in the structured form required for a rule-based type IBMS to process the data easily. An object-oriented format with inheritance allows rules to be written which make decisions on any objects in the same class irrespective of their number and reduces the amount of object description required in the database.

3.2.1 Plant Equipment

The plant equipment database is the largest of the five and contains a record for each item of plant equipment within the IBMS. This data is used for control and maintenance applications. Each item of equipment is represented as an object and has a number of properties associated with it. The exact properties depend on the type of equipment, some of which are plant type specific. For example, a *lamp* may have an average luminaire life property whilst a *motor* may have a phase number property. Other properties are not specific to a particular type of plant such as the description and unit cost properties. Representing this information requires the use of the class and sub-class structure with global properties defined at the parent class level and plant specific properties defined at the sub-class level. Fig. 3.2 illustrates this approach.

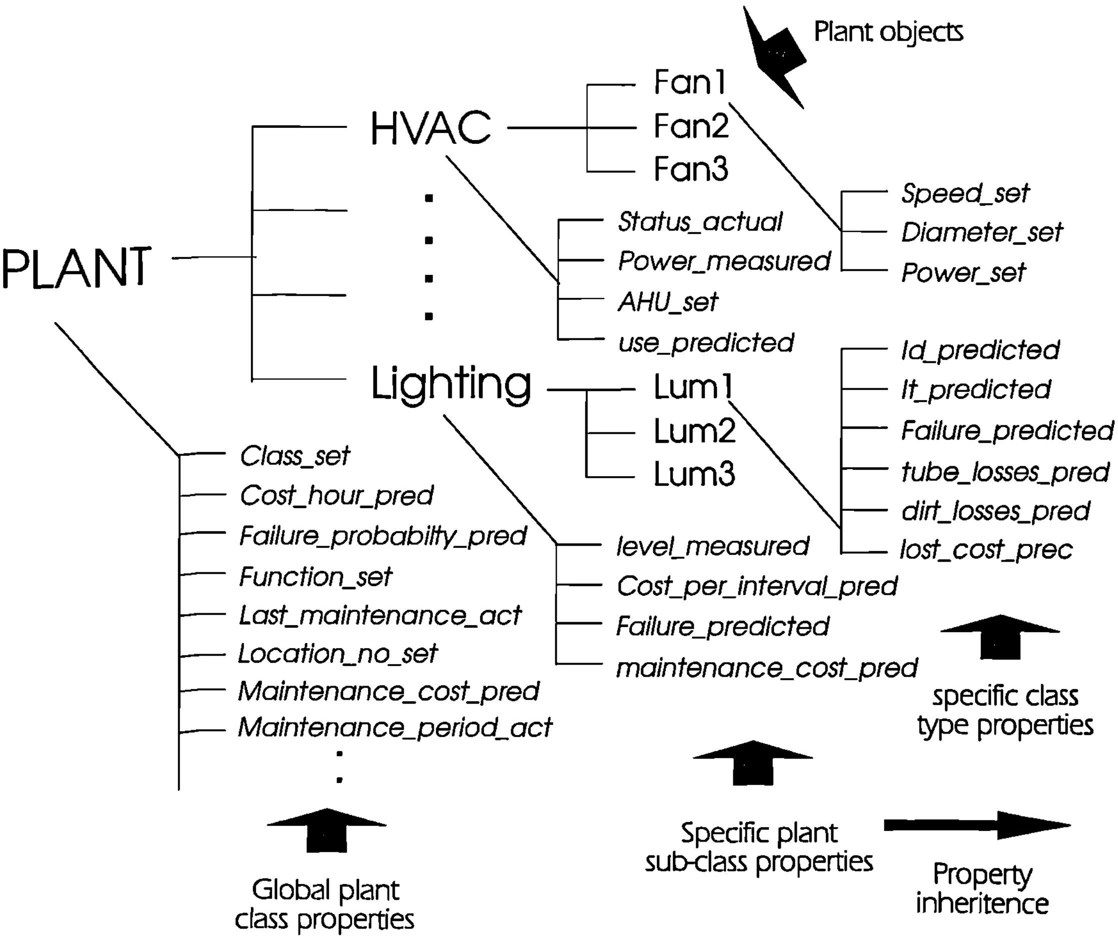


Fig.3.2 Property Inheritance Structure for the Plant Database

The diagram shows the class and sub-class relationship and where different properties are defined. Properties and values can be inherited from classes or objects downwards to another class or object. Inheritance provides efficiency, as the particular attribute only needs to be declared in one place. It also provides consistency as everything which inherits an attribute behaves in the same way. Multiple inheritance is even possible[9]. The properties defined for the parent class of plant equipment are inherited to all objects, these include:

- The class of equipment e.g. HVAC, Lighting, Electrical, etc.
- The plant identification number
- A description of the function of the equipment
- The date of last maintenance (planned or breakdown)
- The cost of planned maintenance
- The planned maintenance period
- The monthly running cost
- The location within the building
- The area(s) within the building which the plant influences
- The unit cost

Much of this data is used by the maintenance management rule-base and is described in more detail in chapter five. Individual sub-classes may have specific properties. For example, the lighting class has the additional specific properties of light level and the HVAC class has a property which stores the number of the associated air handling unit for the equipment. Finally, there may exist specific object properties which may form a further sub-class. In the case of HVAC, a fan may have a property of diameter or speed; a type of fluorescent tube may have specific light level depreciation constants.

When an item of plant occurs many times in the building its properties are defined as a plant type class. For example, there are several hundred carbon bag filters in the air handling units at Lloyd's. It would be inefficient use of computer memory to store a description for each item of plant. Therefore a plant description class is created for the bag filter and properties are inherited from this to the objects. This makes use of multiple inheritance as properties are inherited to the filter objects from more than one parent class.

---

[9] Multiple inheritence in when an object can inherit properties from more than one class.

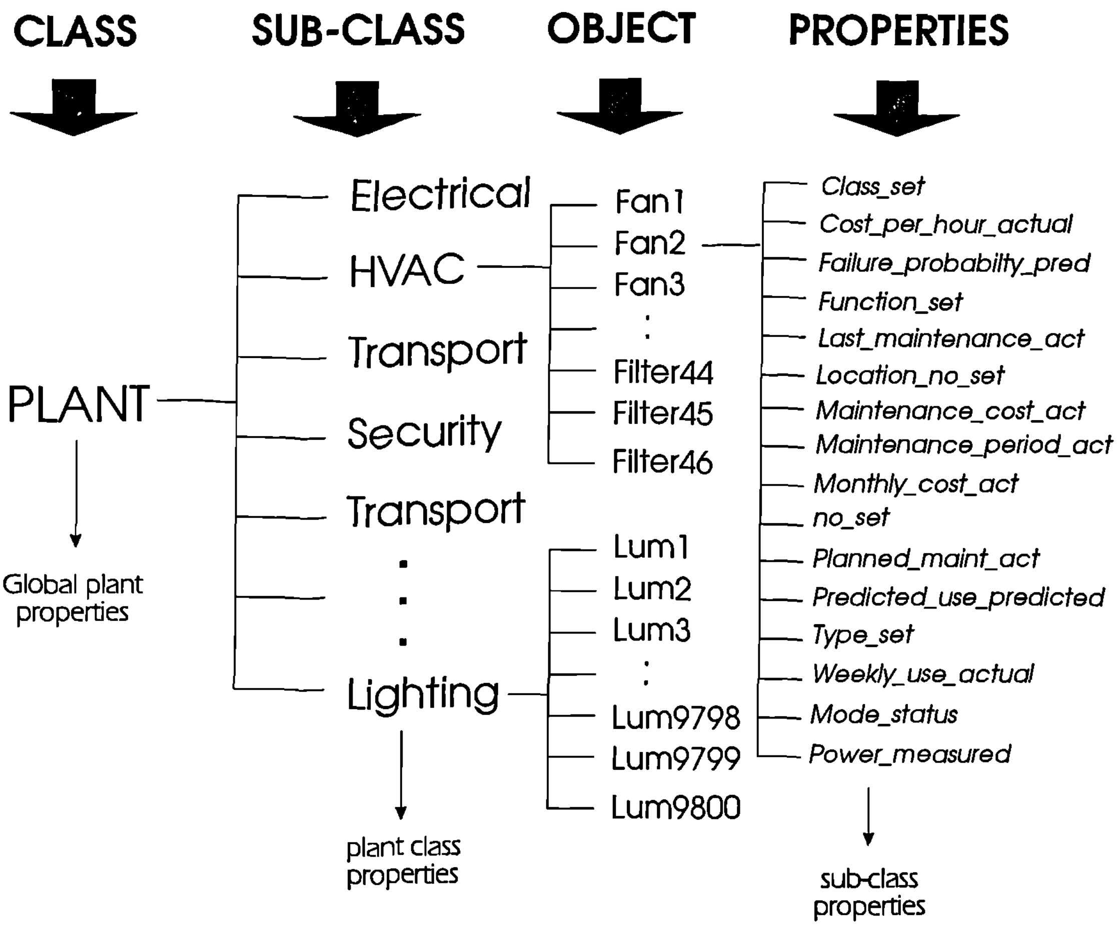


Fig.3.3 Plant Data Representation Structure

Fig.3.3 illustrates the breadth of the plant data structure showing the properties inherited to the sub-class of air handling unit fans. The case studies which illustrate the rule-based approach proposed in this thesis only make use of the electrical, HVAC and lighting databases using data collected from the Lloyd's CALMS and ECS computer systems during the spring of 1991. The lighting database includes data from the 9800 luminaires throughout the twelve galleries in the building. The HVAC database contains information on 750 items of equipment throughout the building. These include air handling units, fans, pumps, filters, boilers, coolers, chillers, pressure units and humidifiers. It is not a full asset register of the Lloyd's plant equipment but includes a broad range of equipment from all main sub-classes. All items of plant used for the building as a whole, such as the main boilers, were included. Where plant items are duplicated throughout the building, data was included only for floor four. In this way the data is as general as possible without becoming unnecessarily repetitive. More information on the data stored within the plant equipment database is given in Appendix C.

3.2.2 Maintenance Database

The maintenance database consists of two separate classes of data. There is a breakdown class relating to information on all reported faults and the fault class, which stores information on cumulative fault statistics. The breakdown data is used to update the fault database. The maintenance database assumes that the building has a central fault reporting facility[10]. When a fault occurs a breakdown record is created and various items of information are stored such as the fault type, the repair time and the engineer who conducted the work. Most of this information is entered after the work is done. This is facilitated by the return of a docket by the repairer when the job is completed. Besides providing useful maintenance statistics this provides a means of keeping a record of the time usage for the staff. Chapter five outlines how the historical maintenance data can be used for trend analysis and fault diagnosis. This information can also be used in other ways within an integrated BMS for optimised preventative maintenance scheduling as described in chapter five.

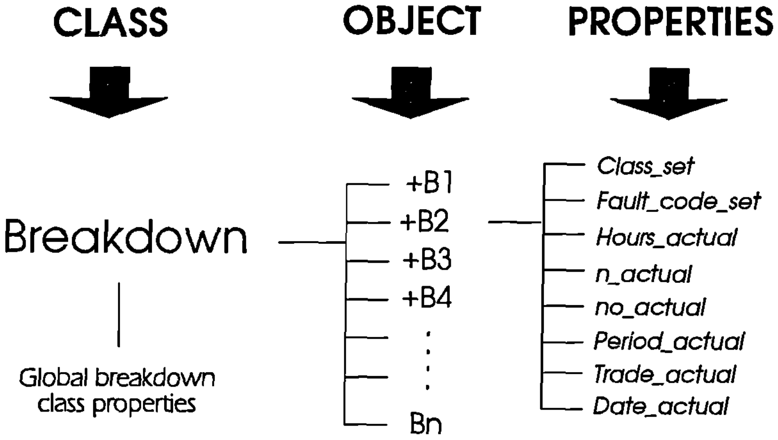


Fig.3.4 Breakdown Data Representation Structure

The breakdown data class and object structure is shown in Fig.3.4. After retrieval from the database the information exists as dynamic objects which are defined solely at run time. This is signified in the diagram by the '+' sign preceding the object's name. A dynamic object is used when the exact objects and their relationships are not known at the time of specifying the system. Dynamic objects allow a knowledge-based system to be general and not specific to any one situation. The objects inherit properties from the main breakdown class such as the fault code, repair time, the fault plant class and the identification number.

---

[10] In the Lloyd's building this is undertaken by the front desk administration (FDA) who receive all details of breakdowns and problems from the occupants of the building.

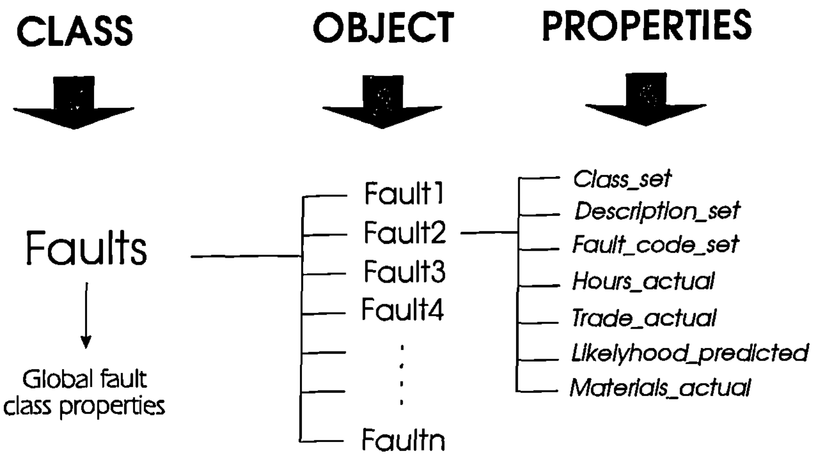


Fig.3.5 Fault Data Representation

A new record is added to the fault description database, shown in Fig.3.5, when a fault occurs for the first time. These are read into dynamic objects and used by the maintenance and fault diagnosis systems. The fault objects inherit all their properties from the main class. Each fault is specific to one or more classes and has a text description explaining the problem in plain English. Other properties include the average value for the fault repair time and the predicted breakdown probability for this class. Derivation of these properties is outlined in more detail in chapter five.

3.2.3 Building Database

The building database stores data relating to the design and operation of the building. Like the plant database it is structured into two sections; one storing fixed data and one dynamic. Fixed data includes properties such as the insulation characteristics of the walls and the room sizes. This dynamic data is updated by sensors in the building and takes the form of occupancy, temperature, fire sensor data and the security status. The building data is represented in the class, sub-class and object structure as shown in Fig.3.6.

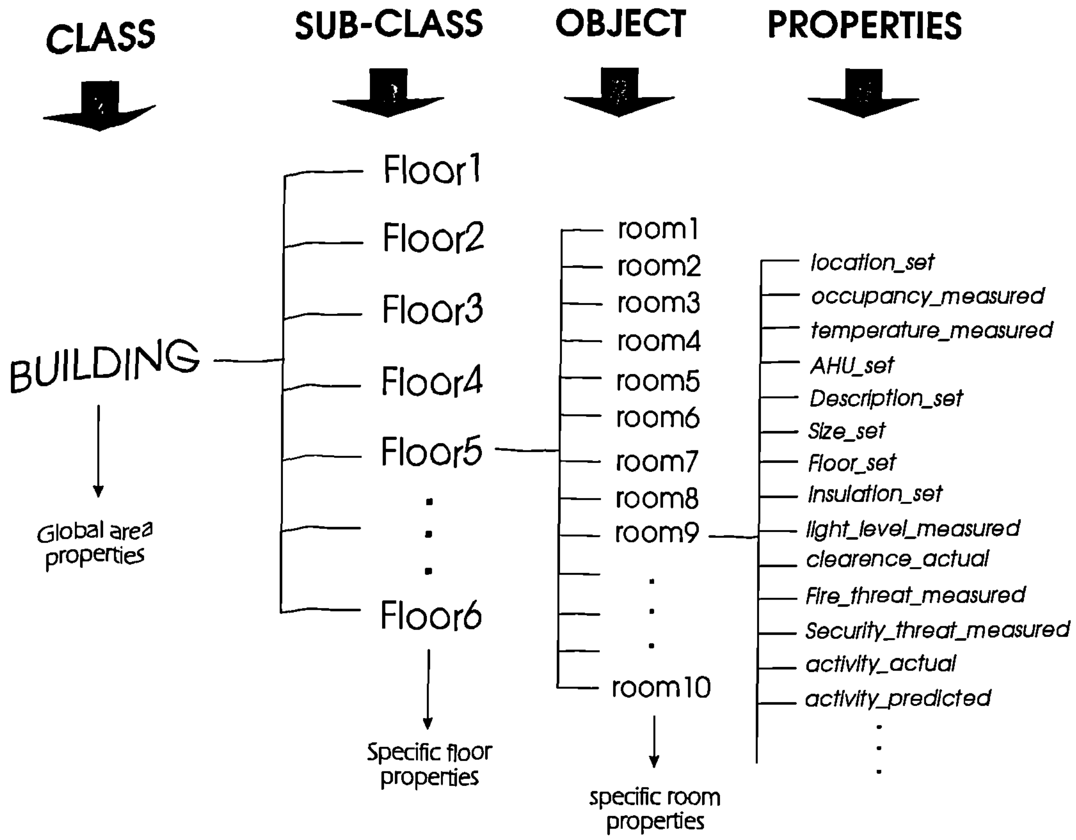


Fig.3.6 Building Data Representation

This assumes that the building is made up of floors each containing a distinct number of zones or rooms. The Lloyd's building is constructed in this way with fifteen floors including two basements. Each room is an object with many of its properties inherited from the main class. In some cases however, some property definition at the sub-class level may be preferable[11]. Although defined as a building with floor and room object representation this need not be the case. The Lloyd's building could alternatively be described as consisting of towers with a central atrium and associated floors as sub-classes. The definition is not important as long as the configuration of the building is represented in the data structure. This then allows the knowledge-base system to reason with knowledge such as which rooms are on the same floor and thus enabling it to control services collectively in related areas.

Although room temperature sensors and in some cases light sensors are common in BMS the adoption of occupancy sensors has still to be widely used. A unique feature of the integrated control methodology is the use of occupancy data and occupancy prediction. The building insulation and volume data are used to model the thermal characteristics of the room for

---

[11] Certain floors may have quite differing characteristics that require special properties. For example, the top floor may have a glass roof and special light and temperature sensors not included on other floors.

optimum start and stop HVAC control. These applications are outlined in more detail in the next chapter.

3.2.4 Resources Database

A building requires resources in many forms in order to function correctly. The 'Resources' database stores this information in three sub-classes which have been defined as 'Trade', 'Energy' and 'Stores'. These sub-classes are used for the knowledge-based applications investigated in the work but do not represent a complete set. The 'Trade' sub-class covers all the maintenance staff available to the estates department for breakdown or preventative maintenance work. The 'Energy' class stores data on all energy used by the building, in whatever form. In the case of Lloyd's, many of the systems are electrical, but there are also oil and gas fuelled boilers. The 'Stores' data keeps a record of all spare part stock levels retained in the building (the assumption is that some stores are kept within the building and that the maintenance is not fully contracted out to an external company). The re-order level for stores is a non trivial calculation for any organisation. Keeping too many spares cost money in terms of storage space and a lack of return on money tied up in stock. Storing too few or no spares can cost money in terms of increased equipment down time.

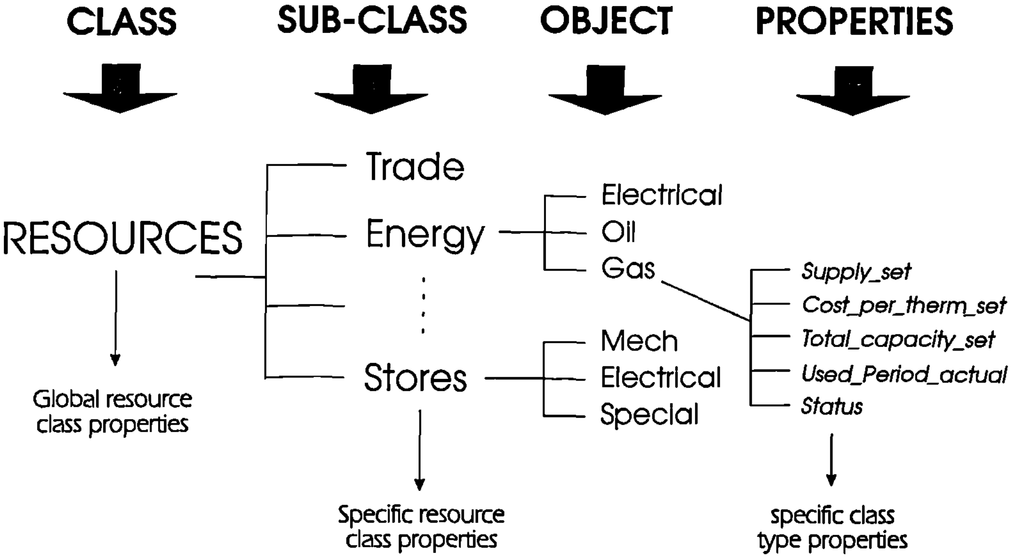


Fig.3.7 Resources Data Representation

Fig.3.7 shows the class and object structure for this data. Similarly to the plant equipment, the class and object properties are inherited from the main class and sub-classes. One of the main properties linking all the objects in this class is that of cost per unit resource. For example, maintenance staff have labour cost and energy is charged per kWh for gas and electricity[12]. One of the prime uses for this data is for evaluating the running costs of the building. For example, the facilities manager could call up a graph of electricity and gas consumption for the last six months or the cost of breakdown maintenance in terms of contract maintenance labour. This has not been investigated in this work as it is implemented already on some of the more advanced BMS systems. Instead, this data has been used in the control and maintenance rule-bases to reduce the operational cost of the building. The next chapter describes how the energy management in the building can be optimised based on the principle of majority person comfort. If there is a power outage, services can be shed based on equipment priority ranking in order to make best use of the available energy. The decision of whether to run the building's own generators for peak lopping is based on offsetting the generator's operation and start-up costs against the savings made in electricity and reduced tariffs. This requires knowledge of relative energy costs. There is a greater incentive for such schemes following the privatisation of electricity generation and distribution in terms of the varying tariffs offered to large consumers at different times of the day.

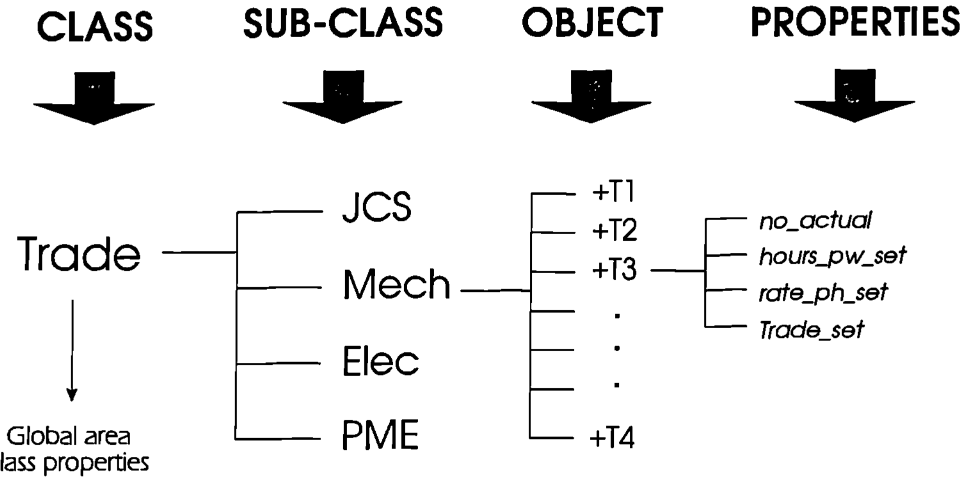


Fig.3.8 Trade Sub-Class Data Representation

Fig.3.8 shows an expansion of the resources data structure for the 'Trade' sub-class. The building's management is likely to employ some of their own maintenance staff in addition to having contract staff and service contracts with their equipment suppliers. The objects shown are dynamic and are retrieved into the system for inference when required. Each object has properties which store information on working hours and cost. The maintenance management section in chapter five outlines how knowledge of the number of maintenance man hours available is used to schedule preventative maintenance, whilst reserving resources for predicted breakdown work.

---

[12] In the UK gas used to be metered in therms but this has recently changed to kWh.

3.2.5 Planned Events Database

The planned events database is the final classification of data. This is used by the control, security and maintenance rule-bases. Each event has a classification, time, date and a text property giving a reason for the particular event. Fig.3.9 shows the class and object structure for the 'Planned events' class.

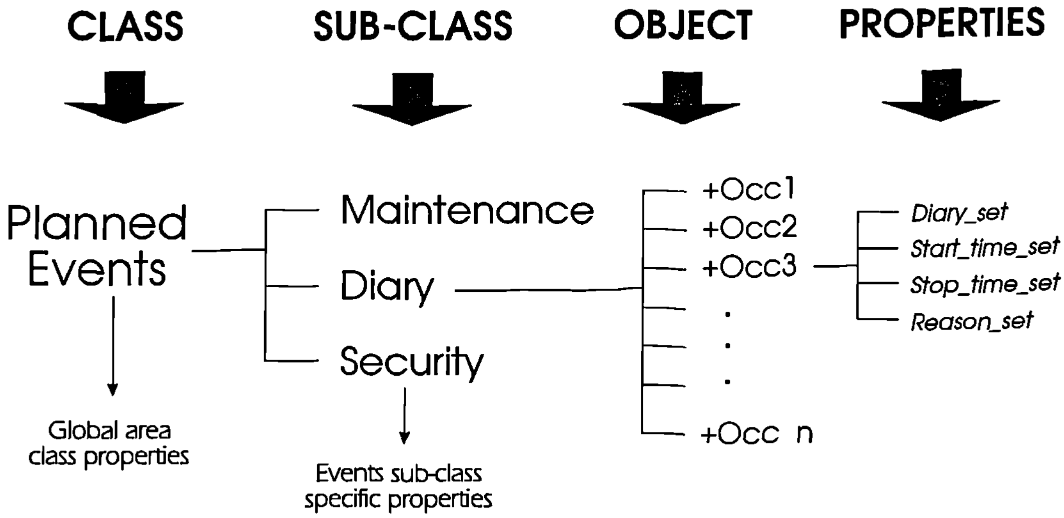


Fig.3.9 Planned Events Class and Object Data Structure

Planned maintenance is classified as a planned event. With this information the IBMS can schedule resources and ensure that the plant and associated equipment are automatically deactivated at the repair time. The building diary database allows the occupants to inform the estates department about the planned usage of the building. Prior knowledge of abnormal out of hours occupation allows the control knowledge-base to schedule services such as lighting and heating. This overrides the zone controller's occupancy monitoring and prediction methodology thus allowing services to be scheduled for out of hours activity. It also allows maintenance to be scheduled at times which cause least inconvenience to the occupants of the building. This information can also be integrated into the security system to explain intruder alarms.

3.3 Rules

The intelligence is built into the IBMS using rules. This section specifies the rule format and classifications adopted by the author. This representation is independent of any particular software implementation.

3.3.1 Rule Description

The IBMS is described in terms of objects. The rules provide the ability to reason on the objects within the knowledge domain. Rules capture the knowledge necessary to solve particular domain problems and include the representation of: relations, heuristics, procedural knowledge and the temporal structure of knowledge. In this thesis the rules are written in the following form.

If condition_A is True
  & condition_B is True
  & condition_C is True
  & etc....
⇒ Hypothesis_X is confirmed
  & do action_1
  & do action_2
  & etc ....

A rule's value depends on the state of its condition clauses:

- If no attempt has been made to evaluate the conditions, then the rule's hypothesis will be unknown
- If all of the conditions are evaluated as true, then the rule is set to true.
- If one of them is false, then the rule's hypothesis will be set to false.

A rule may only have one hypothesis, which is an object with a property of type Boolean[13]. A hypothesis may have many different rules leading to it. In the rule examples given in this thesis the name of the rule's hypotheses all start with a capital letter and are written in italics. This distinguishes them from other objects. The name of the hypothesis describes the action of the rule and where possible related actions for a particular class type are given the same hypothesis name.

Rules are symmetric, i.e. they have no inherent direction in them. This means that the rule can either be processed in the forward direction by forward chaining events or in the backward direction by backward chaining[14] events. The symmetrical property of rules means that there is no need to write one set of forward chaining rules and another set of backward chaining rules. All property values used explicitly in the left hand side conditions, or the right hand side actions of a rule, are called data. Hypotheses which are also data are referred to as sub-goals.

---

[13] A term meaning having two values - true or false, named after George Boole who devised a form of algebra using two state logic.

[14] When satisfying the conditions in a rule requires the execution of the actions of another rule, backward chaining occurs.

Rules can manipulate the values of properties as well as the object and class structures. Pattern matching allows the rules to reference objects which are determined at runtime. Thus one can write generic rules, which reason on a set of objects which are determined when the rule is processed. This allows one to model a world whose exact structure is not known at the time of writing the knowledge-based system. For example, the following condition checks that all the objects belonging to the class 'Hvac' have a power greater than 0.

```
if |Hvac|.power > 0
```

The lines are used to indicate existential pattern matching. If there exist some objects belonging to the class 'Hvac' that meet this criteria, the |Hvac| set of objects is reduced to this set. This can then be operated on in a later condition clause of the same rule. For example,

```
if |Hvac|.power > 0
  & |Hvac|.status = 'on'
```

This finds the set of objects whose *power* property is greater than 0 and whose *status* property is equal to the string 'on'. An independent pattern matching within the same rule is indicated in the notation by using a different number of lines around the class name. The condition clause is independent of the number of objects belonging to the class 'Hvac' or any other property of those objects. Thus the rule can still be applied whether the building has 0, 10 or 200 'Hvac' objects. Pattern matching can also be combined with simple functions. For example,

```
if |Hvac|.power > 0
  & MAX[|Hvac|.cost]
```

This reduces the set of objects belonging to the class 'Hvac' to the objects whose power property is greater than zero and then further reduces the set to the single object whose *cost* property value is a maximum.

If all the conditions on the left-hand side are evaluated to true, the hypothesis is then set to true. The right-hand side actions are only executed if the hypothesis is evaluated to be true. In contrast to the other two parts of a rule, right hand side actions are not required. They are a series of consequences of the rule being fired which are executed as soon as the rule is verified. There may be any number of right hand side actions. Rule actions can be used to reset the value of the rule's hypothesis to 'Unknown' and force the rule to be re-evaluated. This means that the rule continually backward chains to itself until it is proved false.

Every rule has an inference priority which determines the order in which it is processed when forward or backward chaining is not in operation. In this way, rules which diagnose important events such as a fire or loss of power can be given a higher processing order than rules which handle less important situations.

3.3.3 Knowledge-Bases

A collection of rules intended to perform a common function are defined as a knowledge-base. Two main knowledge-bases have been defined for use within a rule-based IBMS. The first is the building control knowledge-base which co-ordinates the control of all the building services. It ensures that the required environment is maintained in all parts of the building, depending on an area's function and activity, for the minimum operating cost. One part of this knowledge-base is processed simultaneously on each of the building's outstations whilst another part is processed on the central control computer. The control knowledge-base is also responsible for handling emergency situations such as the loss of power, or a fire, along with error reporting and condition monitoring. The control knowledge-base is processed continuously and without interaction with the human operator.

The second knowledge-base is concerned with facilities management functions within the building. This knowledge-base requires interaction with the IBMS operator and is processed only when required by the operator. One of the main features of this knowledge-base is to provide on-line fault diagnosis and decision support for breakdowns and the scheduling of maintenance work. Both of these knowledge-bases interact with the classes of data outlined in this section. Within each knowledge-base rules have been defined to perform specific functions as shown in table 3.1.

	Rule Applications	**Building**	**Plant**	**Maintenance**	**Diary**	**Resources**
CONTROL Applications	Zone Control	▨	▨			
	Global Control	▨	▨		▨	
	Energy Management	▨	▨		▨	▨
	Emergency Handling	▨	▨			▨
	Condition Monitoring	▨	▨	▨		
FACILITIES Management Applications	Problem Reporting	▨	▨	▨		
	Fault Diagnosis		▨	▨		
	Maintenance Mgt.		▨	▨	▨	▨
	Cost Reporting	▨	▨	▨	▨	▨

Table 3.1 : Knowledge-base data class interaction

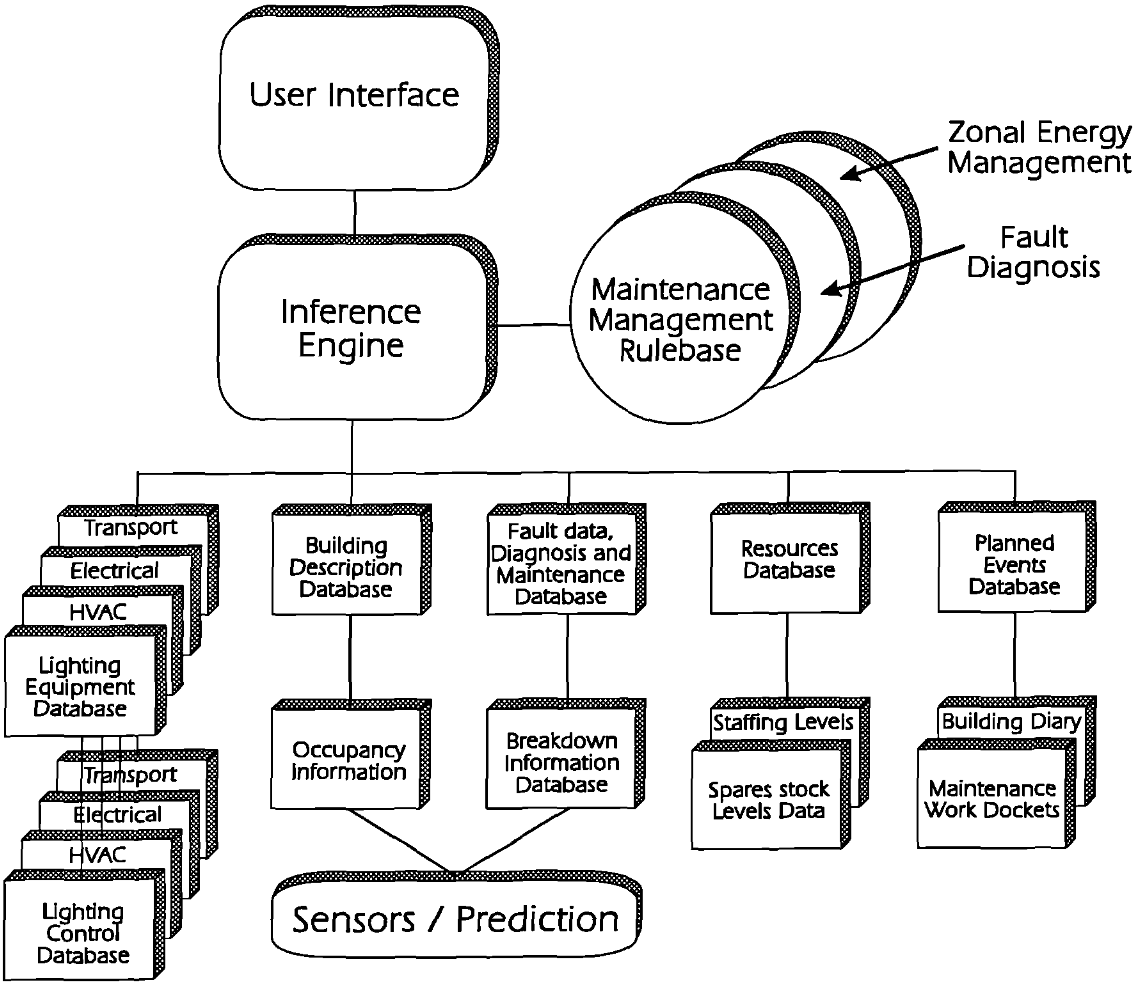


Fig.3.10 Structure for IBMS Rule and Data Integration

The functions are classified by the classes of data on which they operate and in what part of the IBMS they are processed, i.e. the set of rules which only operates on the 'Building' and 'Plant' classes of data are defined as zone control rules. Which rule-base a particular rule belongs to is indicated by the rule's hypothesis name.

Fig.3.10 shows the overall structure for IBMS rule and database integration. The inference engine[15] links the 'intelligence' stored as rules with the knowledge in the form of a database. A prototype system has been built to investigate some of the ideas proposed in this thesis. The specific classes of data required for the KBS have been used to structure an object-oriented database. A prototype has been built within Microsoft's Excel spreadsheet package. Excel is an ideal tool for a prototype database of this nature due to its advanced facility to form dynamic cell links between sheets. The spreadsheet provides a gateway between the data acquisition and the knowledge-base application.

There are many commercial KBS shell[16] software packages available on the market. One such system, Nexpert Object, was used to investigate knowledge-based building management systems [74]. At the start of this work in November 1990, Nexpert was the only development system available which operated under the Windows 3 operating system environment. This was felt to be important, not only because it permitted easy integration with other windows based applications such as Excel and ToolBook, but because at that time it was becoming clear that the whole PC market was moving towards Windows 3 as a standard GUI environment for personal computer applications. Nexpert supports some advanced features including an object-oriented data structure and the ability to modularise knowledge-bases and define inference priorities within rules. Both Nexpert and Excel run within the Microsoft Windows 3.x environment. They both support the Windows 3 direct data link (DDL) feature as well as SYLK database retrieval format which is a derivative of the structured query language (SQL) standard[17]. Nexpert can automatically read in an Excel file from disk and use it to generate an object-oriented data structure, which is then processed by the rule-base.

The next two chapters show how the a data and knowledge structure can be used to implement a rule-based type IBMS by describing the design methodology for the control and facilities management knowledge-bases.

---

[15] The inference engine governs the way in which these rules are processed. This may permit forward and backward chaining as well as special data representation formats. If when the actions of the rule are executed, and it is still not sufficient to fire a rule, the engine must be able to undo assignments of values to variables which prove unproductive and then go on to explore other alternatives. All these features form what is called the inference engine.

[16] An expert system with an inference engine but no rules is called a shell. An expert system shell allows one to develop prototype solutions rapidly.

[17] A standard for data access for many relational data bases like Oracle, Sybase, DB3 and SQL/DS

CHAPTER FOUR

RULE-BASED CONTROL APPLICATIONS

4.0 Overview

This chapter outlines some areas where artificial intelligence techniques, in the form of rule-based systems, can enhance the control operations within integrated building management systems. The aim is to 'tune' the building's services to best match the requirements of the occupants for minimal operating cost.

A hierarchical approach is proposed in which an area's environmental control is achieved by energy management rules running on intelligent outstations. The central IBMS control workstation runs global control rules which have the ability to override the outstation control. Occupancy monitoring and prediction are used in conjunction with the rules which make intelligent decisions on the control of HVAC and lighting. The ideas are illustrated using case studies for both zone and global control functions.

4.1 Rule-Based Control Structure

4.1.1 Structure Overview

An IBMS offers the scope for some advanced control strategies. More information is available, not only on the system under control but also on related systems, which can be used for integrated control functions. This information is used to provide improved energy management and emergency handling features. Three distinct areas of control have been identified:

(1) Global control strategies
(2) Zone environmental control
(3) Local plant control loops

These three control levels, which are defined overleaf, are illustrated in Fig.4.1.

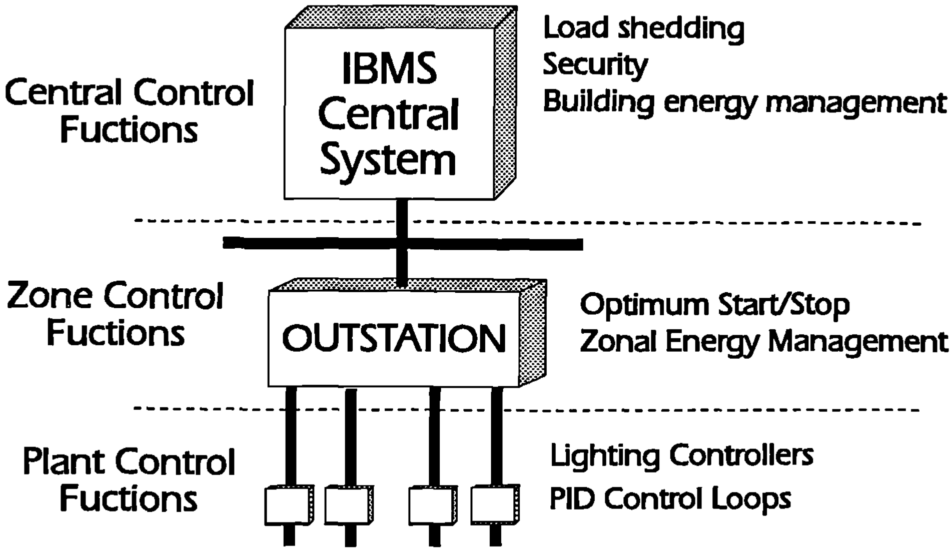


Fig.4.1 Global and Zone level Control Functions

The plant is controlled by local controllers consisting of relays or power electronic switches. Examples of these include PID control loops for heating and thyristor phase angle controllers for lighting. The zone control strategy has the authority to override the PID control loops and ensure the optimal environment for the occupants of the building. A local zone controller is proposed to provide more intelligent environmental control than provided by current systems by tuning the environmental control to the exact usage of the area. This control level uses information from the central IBMS control system along with data from the local sensors. Similarly, when required, the global control level strategies can override the zone control. This enables the IBMS to handle emergency situations such as load shedding and building wide energy management.

Within the context of a knowledge-based IBMS a rule-based control system has been investigated which implements (1) and (2) on the previous page whilst integrating with the existing lower level autonomous control. Several rule-bases have been developed for zone and global energy management within the building and a prototype zone controller was designed and evaluated.

4.1.2 Global Control

Some global control functions would not be practicable in a non integrated building management system. Such applications include energy management in the form of load control and emergency situation handling such as load shedding. Existing BMS control functions include algorithms for load management such as load cycling, load balancing and demand limiting [75]. The rule-based approach does not try to replace these functions but rather to enhance them by optimising their performance.

Recent changes in local electricity metering will mean that the IBMS will have more information on energy consumption throughout the building. Since the privatisation of the electricity supply industry, which split it into three distinct activities (generation, transmission and supply), the element of competition has been increased. Large consumers with a maximum demand greater than 1MW can buy their electricity from any licensed supplier; this is called second-tier supply. This increases competition and is thus an incentive for increased efficiency and price control. The government's plan is that all consumers should have the option of second-tier supply by April 1998 [76]. The minimum requirements for this change are in the metering technology and the methods for meter readings. At the very least all consumers will need a meter which is able to record and store readings. This may need to be done as frequently as every half hour as the pool[18] prices fluctuate from one half hour to the next. The introduction of this increased metering would enable an IBMS to know what items of plant are consuming at any time. This would permit a building management system to have far more control over energy management with electricity load control.

As the cost of electricity can vary very significantly over a 24 hour period there are clear benefits in being able to limit demand when generating costs are high. With direct control of suitable loads, such as heating and lighting, a rule-based system can reduce peak costs by temporarily shutting down loads to reduce consumption. A similar idea called radio tele-switching has been used for many years by some electricity suppliers. This provides a supplier-to-meter control signal carried on the BBC's long-wave radio transmission by phase modulation. At present this is used to stagger the operating times for water and night-storage heaters using cheap-rate electricity for around half a million customers.

Besides reducing the running costs, the building control system must be able to handle the possibility of supply failure. This actually happens more often than one would suppose. There are certain loads in the building that should not be interrupted such as sensitive electrical equipment and emergency lighting. Provision is usually made for such loads using uninterruptible power supplies (UPS) and standby generators. For example, in the Lloyd's building if the main incoming 11kV power supply is lost all essential loads are immediately transferred to the 250kW UPS system whilst the two 1.75MW standby generators are started. These generators take about 20 seconds to get up to speed. All non essential loads must be shed until the total load demand can be supplied by the available power supply capacity. This is conducted by the Johnson's BMS and supervised by the control room operator.

---

[18] The UK electricity pool is an artificial market into which the generators sell their electricity and the supply companies buy it; the pool is analogous to the stock market.

Methodologies exist for optimum load shedding in large power systems [77,78,79]. However, in building management systems the actions to be followed in the event of an outage are usually pre-programmed into the BMS at the commissioning stage. Individual loads are often allocated a set priority level and when a power outage occurs loads are shed in order of least priority. In reality, the best loads to be shed will depend on the time of day and what is happening in the building. What is required is an intelligent control system which strategically prioritises the loads in the building for optimum occupant comfort during supply outage situations.

Instead of just using pre-programmed load priorities this work has investigated the use of a dynamic system of priorities which are based on many factors such as area usage, occupancy, time of day and real time environmental conditions. These priorities, which are set by rules running on the central control system, make use of information gathered from outstations throughout the building and communicated via the building's data-bus.

4.1.3 Zone Control

The zone controller controls the environmental conditions for a specific area in a more intelligent way than conventional systems. The temperature control diagram for a room under the control of a conventional heating system is shown in Fig.4.2 and Fig.4.3. During the occupied times the temperature is controlled at round 20 degrees within dead band of approximately 5%. During unoccupied times this is relaxed to a lower temperature, which is just sufficient to prevent the area freezing. The energy management system calculates the optimum start time such that the area is at the correct temperature at the beginning of the occupied period. This is based on the outside air temperature, the initial room temperature, the desired comfort temperature, the specific characteristics of the room and the properties of the HVAC system. The optimum start time should be as late as possible for maximum energy conservation. Similarly the optimum stop time should be as early as possible.

There are many buildings in which the occupancy patterns are sporadic or very variable. In conventional systems the occupancy period is programmed into the controller and takes little account of the actual occupancy[19]. An overtime function is often included to prolong occupancy period by one or more hours, or for holidays and recurrent deviations from the normal periods. This is not as effective as using actual occupancy data. Simply linking the system to an occupancy sensor would not be very effective as there is a thermal time delay in setting optimum comfort levels. This would mean that when occupants enter an area it is

---

[19] Some systems have tried using CO_2 monitoring as an indication of occupancy. The response time of a CO_2 sensor system would be too slow for lighting control.

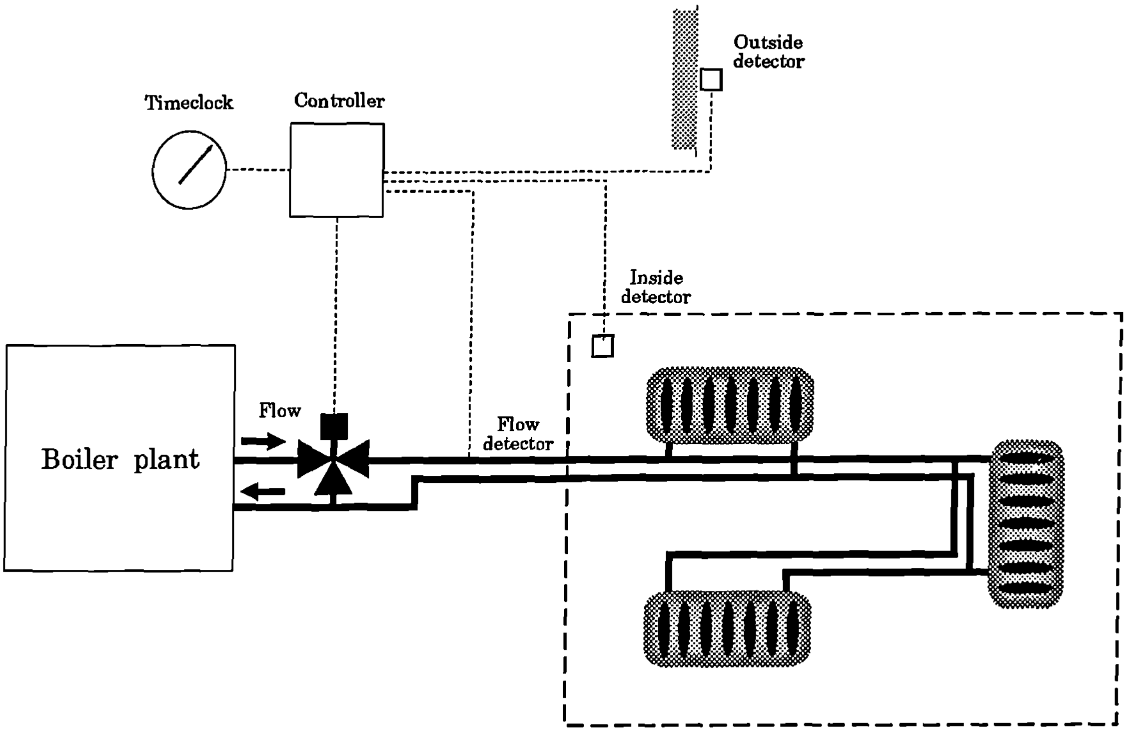


Fig.4.2 Conventional Heating Control System
(Reproduced by kind permission of Brunel University's Estates Department)

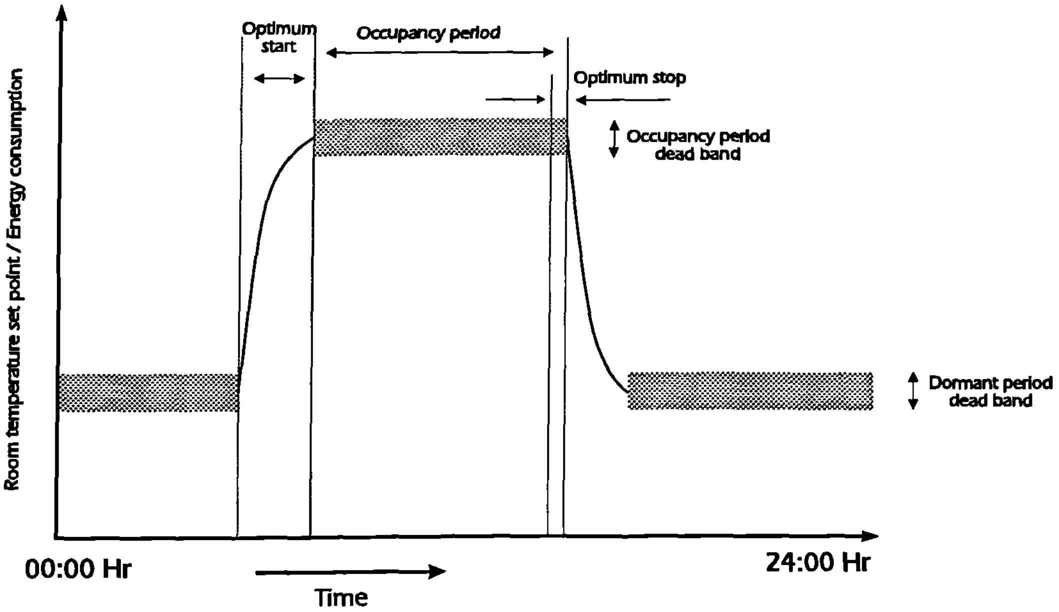


Fig.4.3 Temperature Diagram for HVAC Stop Start Cycles
(after Scheepers, ref.10)

initially cold and the area would remain heated long after occupancy ceased. To overcome this problem requires a zone controller which incorporates intelligent environment control using occupancy prediction in conjunction with people counting. This has been investigated using a rule-based zone controller with rules based on the following knowledge:

- The area's activity
- The predicted activity
- The outside air temperature
- The room's air temperature
- The desired comfort air temperature
- The thermal characteristics of room concerned
- The properties of the heating and cooling system

The rules which engage and control the loads under normal conditions run on the outstations. These rules operate on the properties of objects belonging to the class 'Building'. Actual measured values such as the light level and inside temperature are read in from the sensors and stored in these properties within the controller's memory. The values are then transmitted to the central building data-base when requested by the central control system. Global variables such as the set temperature and light levels are transmitted to the controller by the central control system. In this way the control unit operates autonomously and it is the central controller only that initiates communication[20]. The central controller polls the outstations and updates its building records at regular intervals. The next section describes the rule-based operation of such a zone controller.

4.2 Zone Controller

This section outlines the function of a simple self-tuning rule-based zone energy controller. Rule examples are given for lighting and heating. The rules are equally applicable for the control of cooling systems where rules would take a similar form to the heating rules. The zone control rules for any one individual controller are only concerned with the properties of a single object within the 'Building' class. These will be illustrated using a general object name *ZoneX*, where *ZoneX* is an object belonging to the class 'Building' with properties inherited from that class and its sub-classes. Some of these are measured values, others are set in the building database and some are calculated using methods specific to objects

---

[20] If the controller were to include security and fire detection functions a priority interrupt function would be required to interrupt the main controller and initialte communication. Although not required in the rule base applications outlined in this chapter, it was considered in the design of the outstation hardware outlined in Appendix C.

belonging to the 'building' class. Some of these properties and their sources are illustrated in Fig.4.4.

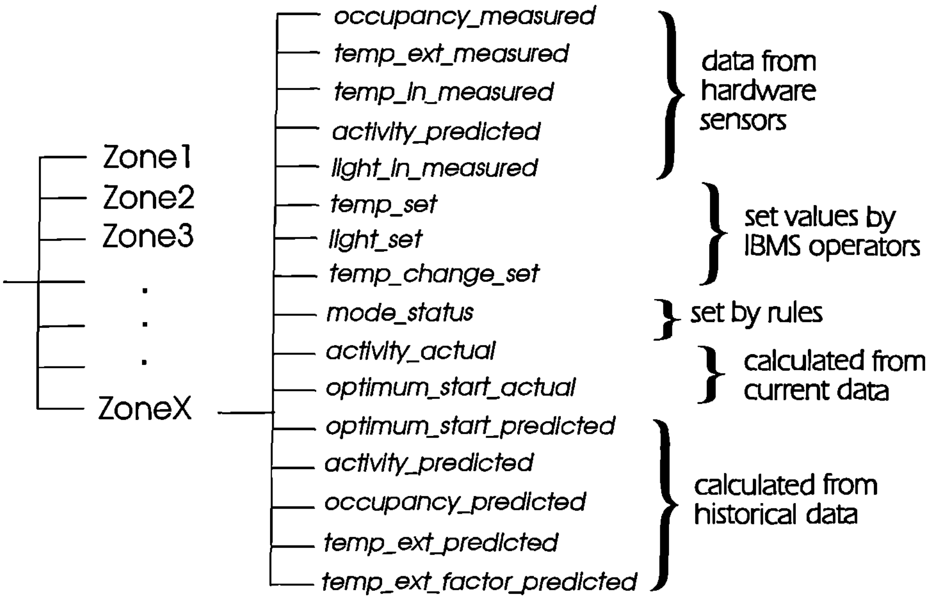


Fig.4.4 Zone Object Properties

4.2.1 Energy Control

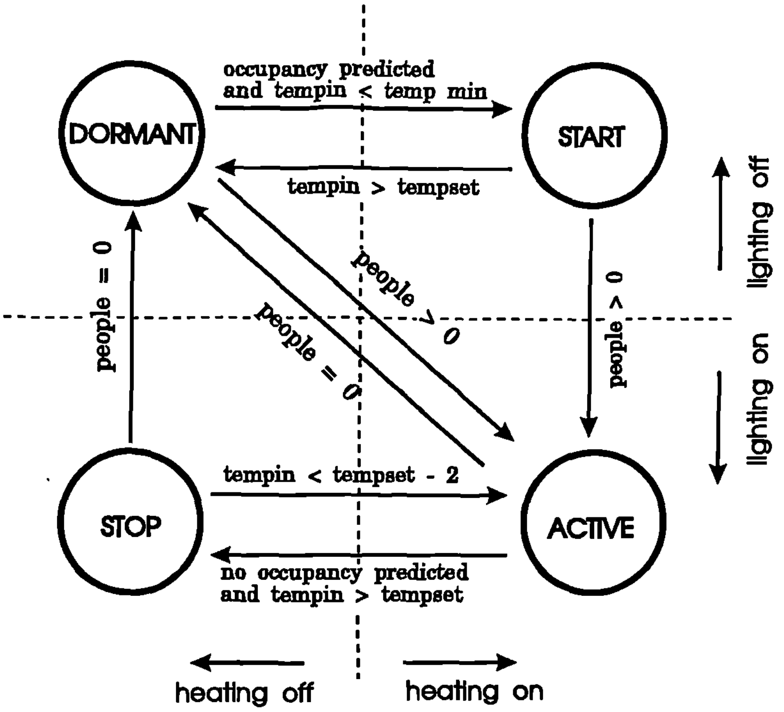


Fig.4.5 State Diagram

The zone control unit operates in one of four states as shown in Fig.4.5. Rules are written to enable the unit to change between these states depending on the occupancy, the temperature, predicted occupancy and information calculated using a mathematical thermal model. The possible state changes are illustrated by arrows on the diagram. The two main states are 'on' and 'off'. In addition to these are the 'stop' and 'start' modes which are entered on the basis of predicted activity. The rules operate on definitions stored in the *activity_actual* and *activity_predicted* properties.

In the 'on' state the area is assumed to be occupied and the temperature and lighting is maintained at the desired comfort level. The heating is maintained by PID controllers and the lighting level is reduced if the external light levels permit. If the area is thought to be unoccupied, the controller switches to an 'off' state. In the 'off' state the area is assumed to be inactive and therefore the heating and lighting are turned off. From the 'off' mode the controller can switch to the 'start' mode using the following rule:

```
If   ZoneX.mode_status is 'off'
     & ZoneX.activity_actual_predicted is 'becoming active'
     & ZoneX.temp_in_measured < ZoneX.temp_set - ZoneX.temp_change_set
⇒    Zone_control is confirmed
     & 'start' is assigned to ZoneX.mode_status
```

If the area is predicted to be becoming occupied the controller switches to the start mode. In this mode the heating is started and switched to full output to keep the pre-heat time as short as possible. The rule ensures that the current room temperature is below the minimal set temperature. This is calculated by subtracting the maximum allowable temperature change from the set point. If the area does not become occupied by the time the desired temperature is reached the controller switches back to its 'off' mode and waits for the area to cool below the minimum set point. The controller switches to the stop mode if the area is predicted to be becoming dormant using the following rule:

```
If   ZoneX.mode_status is 'on'
     & ZoneX.activity_predicted is 'becoming dormant'
     & ZoneX.temp_in_measured ≥  ZoneX.temp_set
⇒    Zone_control is confirmed
     & 'stop' is assigned to ZoneX.mode_status
```

The rule ensures that the current temperature is greater than or equal to the set temperature. If the minimal set temperature is exceeded in the stop mode and the zone is still occupied, a rule causes a switch back to the active mode thus overriding the optimum stop function.

```
if   ZoneX.mode_status is 'stop'
     & ZoneX.temp_in_measured < ZoneX.temp_set - ZoneX.temp_change_set
     & ZoneX.occupancy_measured > 0
⇒    Zone_control is confirmed
     & 'on' is assigned to ZoneX.mode_status
```

If the area's activity is defined as dormant, the controller switches to the shut down mode. A check is first made to ensure that the controller is not in the 'start' mode where no occupancy is allowed.

```
If   ZoneX.mode_status is not 'start'
     & ZoneX.activity_actual is 'dormant'
⇒    Zone_control is confirmed
     & 'off' is assigned to ZoneX.mode_status
```

If there is unexpected activity, rules also exist to jump directly between the 'on' and 'off' states. However, if this activity is known in advance, the 'start' mode can be set from the central IBMS computer and this overrides its own memory.

A fifth state exists called the 'out' state. The controller incorporates no rules to jump into or out of this state. This is used by the global control system to disable the operation of the zone controller by temporarily placing it in this state during load shedding operations. This mode is required during an outage as available power is rationed depending on the relative proprieties of all the loads throughout the building. A zone controller cannot access the data on other areas of the building and hence is incapable of making these decisions itself during and outage. Control is thus handled by the central control system.

4.2.2 Occupancy Prediction

For the prototype controller, occupancy levels were stored in the controller's memory at fifteen minute intervals with a moving window of twenty-one days. Prediction is achieved by looking up the value for the time to be predicted based on average occupancy for the previous three weeks at that time and day. If the prediction time does not correspond with a stored value it can be calculated by means of statistical interpolation. This prevents the prediction from being misled by a one off change in occupancy. If the trend continues, the predictor will adapt after three weeks. For example, if an area is always occupied from 9am to 5pm and the use suddenly changed to 8am to 4pm, the predictor would fully adapt to this change after three weeks but a single deviation from the normal pattern would not cause any change. The adaptation time can be set to suit the functionality of the area under control.

The interpolation uses a standard mathematical interpolation formula which operates on values stored in the look-up table. If $f(x)$ is defined as the mathematical function to predict occupancy at time x, in the table the function $f(x)$ is given numerically for a large number of arguments at equal intervals. The table can be used to interpolate occupancy directly or inversely, that is to compute f(x) at a non-tabular value of x, or to find the non-tabular x for which $f(x)$ has a prescribed value. The Gauss formula in conjunction with a difference table is then used for interpolation at some point between where x_0 and x_1 are two time periods at which occupancy values are stored. This formula is given below. It is one of many similar forms of interpolation such as the Bessel, Neville and Newton forward difference formulas [80].

$$f(x_0+\rho h)=f_0+\rho\delta_{\frac{1}{2}}+\frac{\rho(\rho-1)}{2!}\delta_0^2+\frac{\rho(\rho^2-1^2)}{3!}\delta_{\frac{1}{2}}^3+\frac{\rho(\rho^2-1^2)(\rho-2)}{4!}\delta_0^4+\ldots.$$

Where x_0 is the closest time for which there is a known value, h is the time increment (in this case 20 minutes) and ρ is the fraction of h which when added to x_0 gives the required occupancy prediction time. Truncation after the nth difference gives the value of the polynomial that passes through the n+1 points f_0, f_1, f_{-1}, f_2, etc. The values of δ can be found from the differences between the values of f, i.e. $\delta_0=f_1-f_0$, $\delta_{\frac{1}{2}}=f_0-f_{-1}$, and $\delta_{\frac{1}{2}}^2=\delta_0-\delta_{\frac{1}{2}}$, etc. The software implementation of the occupancy predictor is given in Appendix D and further rules exist in the zone controller to categorise the occupancy.

4.2.3 The Thermal Model

The rules use a mathematical thermal model to calculate the optimum pre-heat and cooling times for the heating control. In practice, this is determined by trial-and-error or look up tables. In the prototype system it is mathematically modelled and the model is integrated within the rule-based system. The optimum start function calculates the latest allowable start time for the HVAC system being controlled, thereby ensuring that the comfort level has just been reached at the beginning of the occupancy period. The start time must take into account:

- The properties of heating and cooling system
- The thermal characteristics of the room
- The outside air temperature
- The room air temperature
- The desired comfort limits

The heat energy input for the heating system is determined by the plant size ratio. This is defined as the ratio of available power input to steady state heat loss rate[21]. Short pre-heating times require large plant size ratios in order to reduce the fuel consumption.

---

[21] This is essentially the definition. However, the exact CIBSE definition is the ratio of normal maximum plant output to the design load for 20 degrees rise.

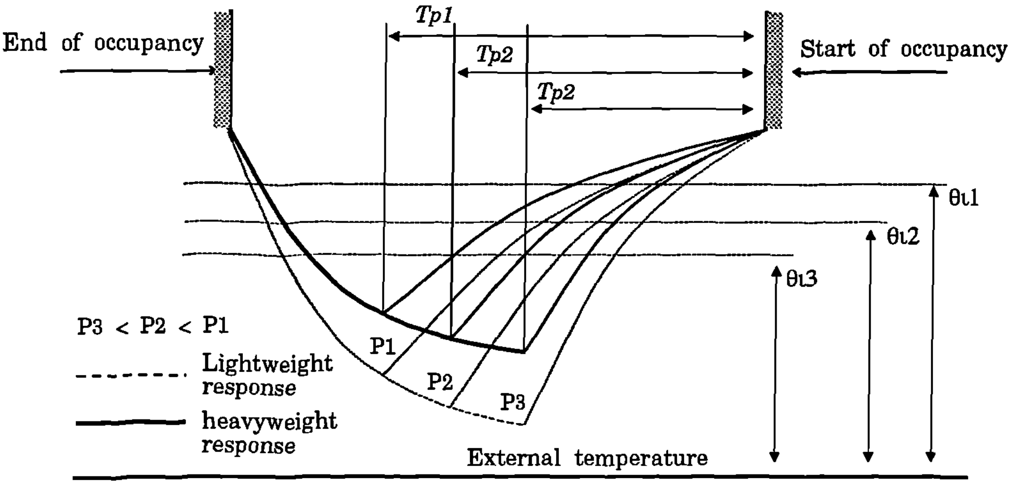


Fig.4.6 Effect of different plant size ratios (P) on the optimum start time (Tp)
(after McLaughlin, ref.82)

Fig.4.6 shows some hypothetical pre-heating curves for two buildings and demonstrates that short pre-heating time (T_p) requires a large plant size ratio (P) and vice versa. Furthermore, each combination of T_p and P gives rise to a particular average value of internal air temperature θ_i. Large values of P give small T_p and θ_i values and hence tend to give lower fuel consumption and costs.

The plant's thermal responses are considered as short or long and the building's response is termed lightweight or heavyweight. A lightweight structure would be one with little or no solid partition and lined with insulating materials. A heavy structure would be masonry or concrete and sub-divided with heavy partitions or floors. The dotted lines show the response P, T_p for a lightweight building with increased thermal losses.

For maximum efficiency, the plant will run at 100% capacity during the pre-heat period to keep this time as short as possible. In mixed air handling systems, return air is used during the start up period as the temperature difference to be overcome is less and the comfort levels will be reached quicker. The required pre-heating time for a particular day depends on the mean external temperature as shown in Fig.4.7. The cooling response curve will also vary with the wind speed and direction.

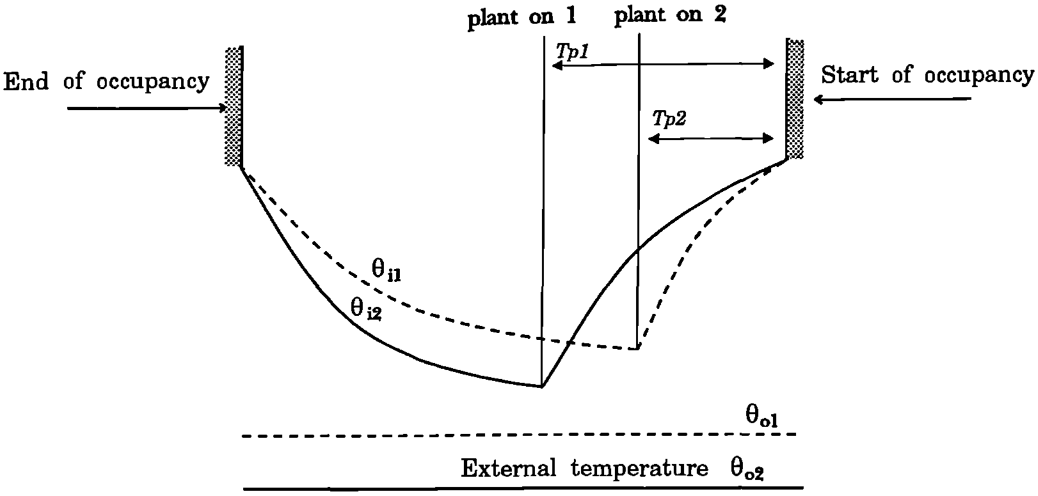


Fig.4.7 Effect of θ_0 on the optimum start time
(after McLaughlin, ref.82)

The following analysis outlines the derivation of a simple mathematical model which is used for the optimum heating start and stop control for a non ventilated room. The following terms are used,

U_f = The overall steady state heat transfer coefficient [$Wm^{-2}K^{-1}$].
A = The surface area over which U_f is applicable [m^2].
Q = Rate of heat flow into room [W].
Q_c = Rate of heat loss through the walls [W].
S_a = Specific Heat Capacity of air [$Jkg^{-1}K^{-1}$].
S_s = Specific Heat Capacity of the structure [$Jkg^{-1}K^{-1}$].
M = Mass of air in room [kg]
m = Mass of the structure [kg]
ϑ = The internal temperature [°C].
θ_e = The external temperature [°C].
θ_o = The initial room temperature at $t = 0$ seconds [°C].
t = The time to reach the desired internal temperature [s].
V = The volume of air inside the structure [m^3]
n = The number of complete air changes per unit time.

If we assume that any other heat gains or losses are negligible, one can consider the heating system and the room as,

Energy in = Air energy increase + losses through structure + energy stored in structure
+ heat lost by ventilation - solar gains - heat generated internally (lights, people, etc)

The rate of heat transfer by convection may be computed by the following equation[22], which can be used to model the losses through the walls.

$$Q_c = A.U_f.(\vartheta - \theta_e)$$

The rate of heat flow due to ventilation is $= n.V.S_a(\vartheta - \theta_e)$

and the heat energy required to raise the temperature of a mass, m, of a substance with specific heat, S_s, from θ_s to ϑ is $= m.S_s(\vartheta - \theta_s)$.

If we consider a room with negligible ventilation, as is likely to be that case during the optimum start period with no occupancy, the energy flow into the room Q in a time Δt which causes a temperature change of $\Delta\vartheta$ this can be expressed mathematically as,

$$Q.\Delta t = S_a.M.\Delta\vartheta + S_s.m.\Delta\vartheta + A.U_f(\vartheta - \theta_e).\Delta t$$

Dividing by Δt and taking the limit as $\Delta t \rightarrow 0$ gives,

$$\Rightarrow Q = (M.S_a + m.S_s)\frac{d\vartheta}{dt} + A.U_f.(\vartheta_i - \theta_e)$$

Which rearranges to give,

$$\Rightarrow Q + A.U_f.\theta_e = (M.S_a + m.S_s).\frac{d\vartheta}{dt} + A.U_f.\vartheta \quad \text{........} \quad (1)$$

Rearranging (1) in a form that can be integrated gives,

$$\Rightarrow \frac{d\vartheta}{dt} = \frac{Q + A.U_f.\theta_e}{M.S_a + m.S_s} - \frac{A.U_f.\vartheta}{M.S_a + m.S_s}$$

$$\Rightarrow \frac{d\vartheta}{\left(\frac{Q + A.U_f.\theta_e}{M.S_a + m.S_s)} - \frac{A.U_f.\vartheta}{M.S_a + m.S_s)}\right)} = dt$$

$$\Rightarrow \int \frac{d\vartheta}{\left(\frac{Q + A.U_f.\theta_e}{M.S_a + m.S_s} - \frac{A.U_f.\vartheta}{M.S_a + m.S_s}\right)} = \int dt \quad \text{........} \quad (2)$$

---

[22] First postulated by Sir Isac Newton (1701)

Integrating both sides of (2) gives,

$$\Rightarrow \frac{\ln\left(\frac{A.U_f.\vartheta}{M.S_a+m.S_s}-\frac{Q+A.U_f.\theta_e}{M.S_a+m.S_s}\right)}{\frac{A.U_f}{M.S_a+m.S_s}}=-t+C \qquad \text{........} \quad (3)$$

The initial conditions before heating begins are used to find the constant C. Assuming that a long period of cooling has occurred and that the thermal structure is at the same temperature as the internal air[23], when $t = 0$, $\vartheta = \theta_o$.

Inserting this into (3) gives,

$$\Rightarrow C=\frac{M.S_a+m.S_s}{A.U_f}.\ln\left(\frac{A.U_f.\theta_o}{M.S_a+m.S_s}-\frac{Q+A.U_f.\theta_e}{M.S_a+m.S_s}\right)$$

Putting this expression for C into (3) gives an equation relating t and ϑ,

$$\Rightarrow \ln\left(\frac{A.U_f.\vartheta}{M.S_a+m.S_s}-\frac{Q+A.U_f.\theta_e}{M.S_a+m.S_s}\right)=-\frac{A.U_f}{M.S_a+m.S_s}t+\ln\left(\frac{A.U_f.\theta_o}{M.S_a+m.S_s}-\frac{Q+A.U_f.\theta_e}{M.S_a+m.S_s}\right)$$

and rearranging for ϑ,

$$\Rightarrow \vartheta=\frac{(M.S_a+m.S_s)}{A.U_f}\left[\frac{Q+A.U_f.\theta_e}{M.S_a+m.S_s}+\left(\frac{A.U_f.\theta_o}{M.S_a+m.S_s}-\frac{Q+A.U_f.\theta_e}{M.S_a+m.S_s}\right)e^{-\frac{A.U_f}{M.S_a+m.S_s}t}\right]$$

$$\Rightarrow \vartheta=\frac{Q+A.U_f.\theta_e}{A.U_f}\left(1-e^{-\frac{A.U_f}{M.S_a+m.S_s}t}\right)+\theta_o.e^{-\frac{A.U_f}{M.S_a+m.S_s}t} \qquad \text{........} \quad (4)$$

Equation (4) shows that the temperature change in the room is governed by a function made up of an exponential rise and an exponential decay function. Both terms have the same time constant which is a function of the thermal characteristics of the room and heating system, i.e.

$$\tau=\frac{M.S_a+m.S_s}{A.U_f}$$

---

[23] In practice this is only an approximation as a significant amount of heat energy may be stored in the structure overnight depending on the cooling time. This fact is allowed for by using more than one thermal time constant as outlined later.

As one would expect, increasing the size of the room and hence the greater the mass of the air or thermal capacity causes an increases in the time constant. Similarly, an increase in the heat transfer coefficient or the wall area causes the time constant to decrease. The exponential rise represents the heat energy input to the room. If the heat supply, Q, is insufficient to overcome the heat losses this could become a negative increasing exponential function. If the heating remains on continuously, the highest temperature the room will reach is given by,

$$\frac{\lim \vartheta}{t \rightarrow \infty} = \frac{Q}{A.U_f} + \theta_e \text{ so if } \theta_r \text{ is the required room temperature, } \frac{Q}{A.U_f} + \theta_e > \theta_r$$

The second part of equation (4) is a simple exponential decay from the initial room temperature. This decays as the room temperature either increases or decreases to a new value.

In an IBMS controller the thermal characteristics will be known and the final temperature will be set by the operator. The pre-heat time will depend on the initial room temperature and the external temperature. Rearranging (4) in terms of time we get,

$$\Rightarrow \frac{A.U_f.\vartheta}{M.S_a + m.S_s} - \frac{Q + A.U_f.\theta_e}{M.S_a + m.S_s} = \left(\frac{A.U_f.\theta_o}{M.S_a + m.S_s} - \frac{Q + A.U_f.\theta_e}{M.S_a + m.S_s} \right) e^{-\frac{A.U_f}{M.S_a + m.S_s} t}$$

$$\Rightarrow e^{-\frac{A.U_f}{M.S_a + m.S_s} t} = \left(\left. \vartheta - \frac{Q + A.U_f.\theta_e}{A.U_f} \middle/ \theta_o - \frac{Q + A.U_f.\theta_e}{A.U_f} \right. \right)$$

$$\Rightarrow t = -\frac{M.S_a + m.S_s}{A.U_f}.\ln \left(\left. \vartheta - \frac{Q}{A.U_f} - \theta_e \middle/ \theta_o - \frac{Q_e}{A.U_f} - \theta_e \right. \right) \qquad \text{........} \quad (5).$$

Equation (5) predicts the optimum pre-heat time for a room given the thermal characteristics, the initial inside temperature and the external temperature. The heat energy stored in the structure will depend on the length of the previous heating or cooling period. This could be measured using sensors inside the structure or predicted using the building's thermal capacity. Buildings exhibit at least two time constants during the heating or cooling phase. For cooling there is an initial rapid drop which characterises the room air and heating plant response which is due to the high U value of windows and the low specific heat capacity of air. This time constant is of primary interest in automatic control during the hours of occupation. A longer time constant is used to characterise the overnight cooling as the structure cools during a weekend shut down or longer periods. Both these values are of importance in optimum start control. Equation 5 assumes that the structure's temperature is the same as the internal air and calculates the time to heat both the structure and the air as might be the case after a long cooling period.

Factors influencing the time constants	Lightweight	Heavyweight
cooling of air and heating plant	30min - 2h	up to 10h
cooling of structure	2-12hr	40-120hr

Table 4.1 Approximate cooling time constants
(after McLaughlin, ref.82)

Table 4.1 gives some typical values but the actual values vary depending on the building's internal surface properties [81]. They are affected by the amount of heat energy stored in the building's structure. Consider the simplified electrical analogy illustrated in Fig.4.8 which assumes a constant external temperature. The system can be modelled by an electrical circuit where the current source represents the heat input, R_u represents the thermal losses through the walls and windows, C_t the thermal capacity of the structure and C_a the thermal capacity of the air. The time constant representing the structure will be longer that of the air. The switch represents the intermittent heating control. If the heating is in daily use, C_t will remain charged and thus the time constant to heat the area (store charge in the capacitors) will be governed by R_aC_a only. To model this correctly more than one time constant is therefore stored in the building data-base for each zone and the rules select the appropriate one depending on the intervening cooling time. The way this function is described in terms of two constants and the self-tuning aspect is covered later in section 4.2.6.

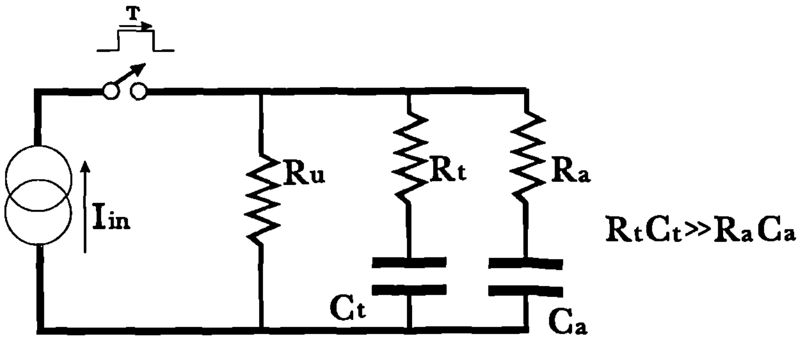


Fig.4.8 Simplified Electrical Analogy

This knowledge is used by the rules to improve the heating and cooling control. For example, Brunel University has quite an advanced Trend energy management system and the heating start and stop times have been set to fit the expected occupancy of the buildings. Last year, during the Christmas holidays all heating was shut down to minimum to reduce running costs during a period when the outside temperature fell to below zero. The fabric of the building was thus allowed to cool for three weeks. The heating system's optimum start controller cut in on the morning of the first day of term on the same day that the cold spell of weather rapidly improved and the outside temperature rose. The energy management system took no account of the three weeks shut down in the optimum start function and as the outside temperature was not too low predicted a relatively short pre-heat time. This resulted in the building being very cold and damp for the next two days and many complaints to the estates department.

Another factor this model takes into account is variations in Q. It cannot be assumed that the building's boiler will give a constant output. This is another problem experienced by the users of Brunel's EMS.

Depending on the length of the unoccupied period, the initial air and structure temperatures may be anything between the desired comfort level and the external temperature[24]. Similarly, the external temperature varies for the time of day and year. Fig.4.9 shows the variation in pre-heat time calculated using equation (5) for a range of initial inside and outside temperatures. In each case the desired internal temperature is taken to be 23 degrees.

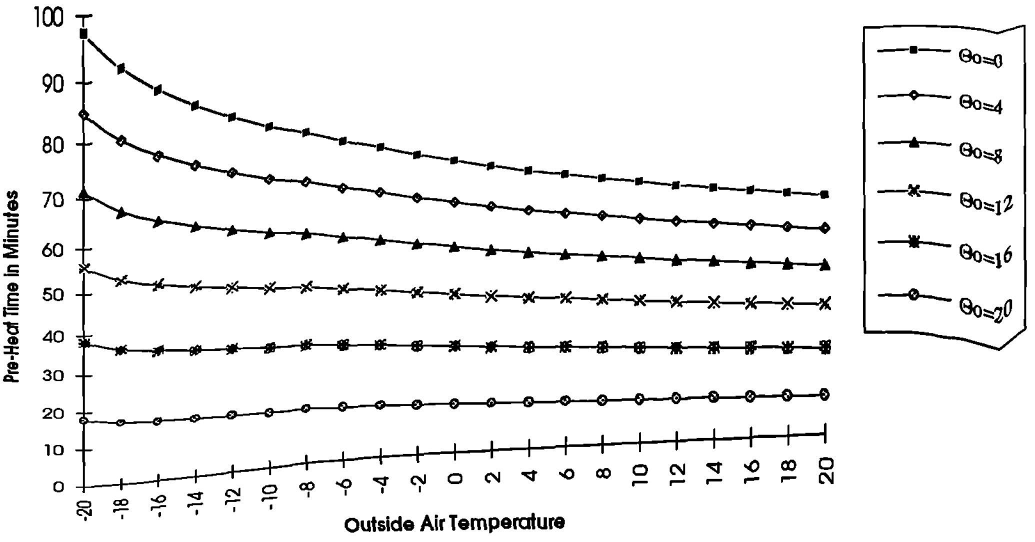


Fig.4.9 Graphical Representation of the Optimum Start Function

---

[24] In practice the minimum temperature will be preset to protect the fabric of the building and its contents. Heating will automatically cut in to maintain this minimum temperature.

The optimum stop function can be found by using equation (5) with $Q = 0$ as shown below,

$$\Rightarrow t_{stop} = -\frac{M.S_a + m.S_s}{A.U_f}.\ln\left(\frac{\vartheta - \theta_e}{\theta_o - \theta_e}\right) \quad \ldots\ldots\ldots \quad (6).$$

The rule-base uses equations 5 and 6 to calculate the optimum preheat time for an area thus ensuring that minimal energy is consumed. The specific constants for each zone are stored in the building data-base along with the current data from the sensors in that area.

4.2.4 Actual Control

Once the rules have come to a decision on the environmental control the heating, cooling and lighting is controlled by passing parameters to the hardware controllers. For the prototype system five control codes have been defined:

Level 0 : the load is switched off
Level 1 : the load is switched fully on
Level 2 : the load is under closed loop control
Level -1 : the load is under closed loop control but with a reduced set point
Level -2 : the load is shut down for maintenance

During the pre-heat time the heat output is usually set at maximum to keep this time as short as possible. This would require a level 1 control for maximum efficiency. Following the pre-heat stage it is usual to maintain the temperature at a set value. This is defined as level 2 control. For heating closed loop control may be achieved by standard PID control loops. These are used to set the position valves or dampers in the case of a wet or dry HVAC system to exactly offset thermal losses or gains. The light level can also be accurately controlled to make maximum use of natural daylight. Fluorescent tubes may be dimmed by incorporating a thyristor controller that varies the current through the tube independent of the heating current. This is illustrated in Fig.4.10

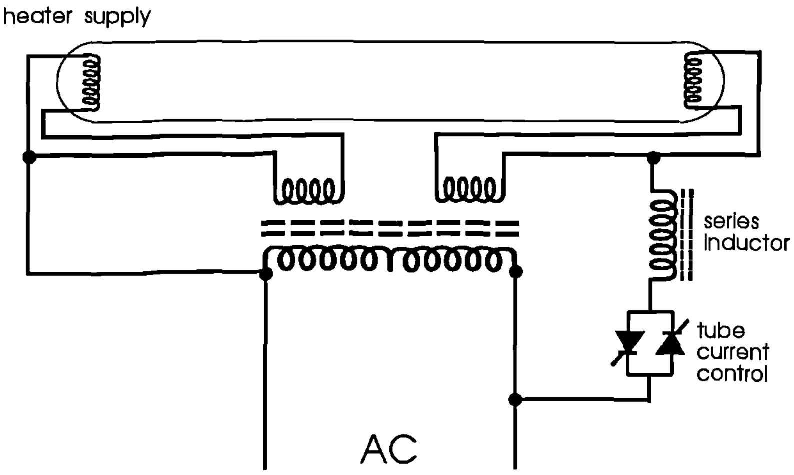


Fig.4.10 Fluorescent Tube Dimming Circuit

This circuit is a variation of the standard quick starter configuration. The light level can be controlled by changing the thyristor firing angle using a standard thyristor controller circuit. As the firing delay angle increases, the light output decreases. In conjunction with an inside light sensor the light level is reduced until the minimum set value is achieved. This minimum value depends on the time of day as well as the area's activity as described later.

The *control_state* properties of objects in the 'PLANT' class are used to pass these control codes to the plant controllers. The values of these properties are also read by the global control and used to update the plant equipment database. Separate rules are used for switching different classes of plant and in each case the current status of the plant is checked.

If the controller is in 'on' mode, the heating and lighting plant is turned on. This is achieved by setting the *control_status* property of the specific plant equipment to a level greater than 0.

```
if   ZoneX.mode_status is 'on', 'start'
     & Hvac_n.control_status $\neq$ 2
     & ZoneX.temp_in_Measured $\geq$ ZoneX.temp_set
⇒    Zone_control is confirmed
     & 2 is assigned to Hvac_n.control_status

if   ZoneX.mode_status is 'on', 'start'
     & Hvac_n.control_status $\neq$ 1
     & ZoneX.temp_in_measured $<$ ZoneX.temp_set
⇒    Zone_control is confirmed
     & 1 is assigned to Hvac_n.control_status

if   ZoneX.mode_status is 'on', 'stop'
     & lighting_n.control_status $=$ 0
⇒    Zone_control is confirmed
     & 2 is assigned to Lighting_n.control_status
```

The second rule is required to prevent saturation of heating PID the control loop if the optimum start time is incorrect and the set temperature is not reached at the start of occupancy. When the controller enters the 'on' mode switching to PID control will cause the control loop to saturate as it tries to correct for the temperature error. This could cause the temperature of the room to increase considerably above the set value and would not be immediately corrected due to the phase shift resulting from the room's thermal time constant. This is called overshoot and wastes energy. The second rule ensures that, should this occur, the level 1 control is maintained until the set temperature is reached. Under level 1 control the integral and derivative actions of the controller are switched off thus preventing saturation.

In the case of lighting and heating equipment rules exists to prevent excessive switching of plant fully on and off. For example, if someone consistently enters and leaves an area, rules ensure that on the second time of leaving the lights remain on until the area has been in the dormant state for at least fifteen minutes. Further short term occupancy extends this switch off delay time. This makes use of the property *lighting_off_time_actual*. For example,

```
if  ZoneX.mode_status = 'off' , 'start'
    & lighting_n.control_status > 0
    & time_measured - ZoneX.lighting_off_time_actual > 900
⇒   Zone_control is confirmed
    & 0 is assigned to Lighting_n.control_status
    & time_measured is assigned to ZoneX.lighting_off_time_actual
```

Similarly, heating is turned off during the 'stop' and 'off' modes. In this way the environmental control is tuned to the activity in the area.

A simulation was conducted in consultation with engineers at Dunwoody and Partners to predict possible energy savings using a building services design package called HevaStar. This is a suite of software packages for building services engineers to use for computer aided design and design evaluation. This package has been widely used by the engineers at Dunwoody for several years. Two programs were used: 'HLOSS' which estimates the building's thermal energy loss and 'ENERGY' which calculates the annual heating energy consumption of the building. This is done by firstly calculating the design heat loss of the building and then by using a version of the degree day method, modified to allow for permissible solar, lighting and people heat gains using techniques from the CIBSE Energy Code Part 2(a). The largest building on the Brunel campus was chosen for the simulation which considered heating, lighting and ventilation savings. The full results are given in Appendix E.

The results of the simulation on tuning the control of the services to occupancy suggest that an annual saving can be made of around 3.7% for a 1 hour reduction in heating and 7.7% for 2 hours from a normal 9 hour occupancy period. The simulation predicts electricity savings up to 20% for the building in question. In order to confirm that the HevaStar predictions were reasonably accurate the predicted running costs for the building in question were compared with the actual running costs shown in Appendix E and found to be close. This information was provided by the estates department of Brunel University. The energy savings will of course depend of the type of room, its occupancy patterns and the time of year. Obviously higher savings can be made in sporadically occupied rooms where HVAC set points can be relaxed for more than an hour. In addition to this, the self-tuning aspect would save a great deal of an estates department's staff's time in re-configuring control schedules when the functions of areas change.

4.2.5 Activity Filter

Rules are used to filter the data from the occupancy sensors and convert it into activity level information for the various areas in the building. The rules are split into two sets namely 'actual activity' and 'future activity'. The operation this filter is illustrated in Fig.4.11. The actual activity will either be 'dormant', 'quiet', 'active' or 'normal'. An area is defined as being dormant if the occupancy level is zero. Quiet areas are occupied, but not as occupied as predicted. Dormant and quiet area definitions are used by the economy rules in the energy management rule-base to effect energy savings in the building. These areas are defined by the load shedding rules as being low priority if there is a power outage. Areas are defined as 'Normal' if the occupancy broadly fits the predicted levels. Finally, active areas are where the occupancy is significantly higher than expected. Active areas could signify that the function of the area is temporarily abnormal and hence the future occupancy prediction may also be in error. An example of this might be a room being used for an unexpected out of hour's meeting.

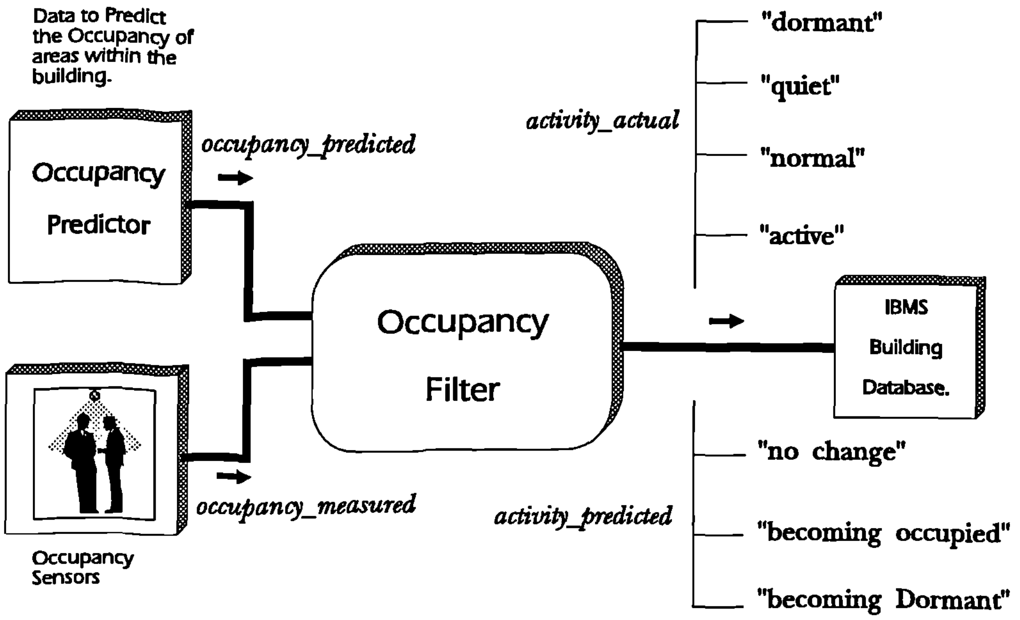


Fig.4.11 Occupancy Filter rule-base

The first rule to be fired defines an area as being dormant. This is used by the controller to switch to the 'off' state and is shown below.

```
if   ZoneX.occupancy_measured = 0
     & ZoneX.occupancy_predicted = 0
⇒    Dormant is confirmed
     & 'dormant' is assigned to ZoneX.activity_actual
```

The rule sets the value of the *activity_actual* property to "dormant" if the actual and predicted occupancy is both zero. The next rules define areas as 'quiet'.

```
If   ZoneX.occupancy_measured < ZoneX.occupancy_predicted × 0.80
⇒    Quiet is confirmed
     & 'quiet' is assigned to ZoneX.activity_actual
```

The rule shown above checks that the actual occupancy is less than 80% of the predicted occupancy, if it is the value of the *activity_actual* property is changed from the string "dormant" to the string "quiet". The next rule assigns the string "quiet" to the *activity_actual* property if the current operating mode is 'on', the actual occupancy is zero and the predicted occupancy is greater than zero.

```
If   ZoneX.mode_status is 'on'
     & ZoneX.occupancy_measured = 0
     & ZoneX.occupancy_predicted > 0          { calculates predicted occupancy }
⇒    Quiet is confirmed
     & 'quiet' is assigned to ZoneX.activity_actual
```

This rule is included to prevent areas from being incorrectly labelled as 'dormant'. For example, the system would be lacking in intelligence if the economy rules shut off lighting and heating in the middle of an occupied period just because everyone had gone next door for 20 minutes for a meeting. Instead, any area that is currently occupied and suddenly becomes dormant, but which is predicted to be occupied, is defined as being 'quiet'. This way the system errs on the side of caution. If the function of a particular area does change and it is dormant at this time every day the occupancy predictor would retrain with time.

Normal areas are defined as those in which the predicted and actual occupancy levels are approximately the same. The next rule defines areas as being 'normal'.

```
If  ZoneX.occupancy_measured > ZoneX.occupancy_predicted × 0.8
    & ZoneX.occupancy_measured < ZoneX.occupancy_predicted × 1.2
⇒   Normal is confirmed
    & 'normal' is assigned to ZoneX.activity_actual
```

The condition clauses check that the occupancy falls within a ±20% range of the predicted occupancy. This allows for some error in the prediction and variation in the occupancy levels. The string "normal" is then assigned by the rule's action to the *activity_actual* property. The next rule defines areas as being active. It is similar to the above rule except the actual activity must be more than 20% greater than the predicted occupancy levels.

```
If  ZoneX.occupancy_measured > ZoneX.occupancy_predicted × 1.2
⇒   Active is confirmed
    & 'Active' is assigned to ZoneX.activity_actual
```

The next set of rules use information on predicted occupancy to predict whether an area's activity will be 'becoming active', 'becoming dormant' or if there will be 'no change'. This is used by the energy management rules for switching to 'start or 'stop' modes for heating and cooling equipment.

The rule shown below uses the thermal model to calculate the optimum stop time and defines the area's predicted activity as 'becoming dormant'. If the occupancy is predicted to fall to zero concurrent with the optimum stop time, the activity prediction is then defined as 'becoming dormant'.

```
If  ZoneX.mode_status is 'on'
    & calculate ZoneX.occupancy_predicted at time_measured
    + ZoneX.optimum_stop_predicted
    & ZoneX.occupancy_predicted = 0
⇒   Becoming_dormant is confirmed
    & 'becoming dormant' is assigned to ZoneX.activity_predicted
```

When the *optimum_stop_predicted* and *occupancy_predicted* properties require evaluating they call on methods to calculate their current values. The implementation of these methods is outlined in Appendix B.

The next rules are used to determine if the area is currently in an occupied state when the future activity is 'no change'. The rule shown below defines the *activity_predicted* as 'no change'. Whether the actual activity is 'active', 'normal' or 'quiet' is not important it must not be 'becoming dormant'.

If ZoneX.mode_status is 'on'
& calculate ZoneX.occupancy_predicted at time_measured
+ ZoneX.optimum_stop_predicted
& ZoneX.occupancy_predicted > 0
⇒ *No_change is confirmed*
& 'no change' is assigned to ZoneX.activity_predicted

The rule checks that the area will still be occupied at a time in the future equal to the current calculated optimum stop time. The next rule shown defines an area as no change if the *activity_actual* is 'dormant' and the predicted occupancy is zero.

If ZoneX.activity_actual is 'dormant'
& ZoneX.occupancy_predicted = 0
⇒ *No_change is confirmed*
& 'no change' is assigned to ZoneX.activity_predicted

Areas of the building which are currently defined as active and whose predicted occupancy is low enough to define their predicted activity as 'becoming dormant' are also defined as no change by the following rule.

If ZoneX.activity_actual is 'active'
& ZoneX.activity_predicted is 'becoming dormant'
⇒ *No_change is confirmed*
& 'no change' is assigned to ZoneX.activity_predicted

This is included because the active definition assumes that the area is being used for an abnormal function and hence the future occupancy prediction is likely to be in error. When the area's actual activity returns to 'normal' or 'quiet' the *activity_predicted* will then be defined as 'becoming dormant' by a previous rule.

The final rule in this set assigns 'becoming occupied' to the property *activity_predicted* for an area which is currently in the 'off' mode and its predicted occupancy at the current time plus the optimum start time is predicted to be greater than zero. This rule again uses the thermal model, in this case to calculate the optimum start time.

```
If   ZoneX.mode_status is 'off'
     & calculate ZoneX.occupancy_predicted at time_measured
       + ZoneX.optimum_start_predicted
     & ZoneX.occupancy_predicted > 0
⇒    Becoming_occupied is confirmed
     'becoming occupied' is assigned to ZoneX.activity_predicted
```

4.2.6 Self-Tuning Thermal Model

In many systems the pre-heat time is set at the commissioning stage and takes no account of variations in the environment. To improve on this, further rules have been written to enable the zone heating system to become self-tuning. If the actual pre-heat time is significantly different from the predicted time, the system uses the error to firstly diagnose what may have caused the discrepancy and then secondly to re-tune the timing to reduce such errors in future. If the predicted pre-heat time is incorrect, assuming a good model of the system, the error could be due to one of the environmental constants having changed or the outside temperature changing during the pre-heat phase. Optimum start controllers are usually built to start the plant on the assumption that θ_e will be constant during the pre-heating period. In general this is a conservative estimate and leads to an early attainment of the desired internal conditions. This reduces efficiency by unnecessary fuel consumption. Fig.4.12 illustrates this point.

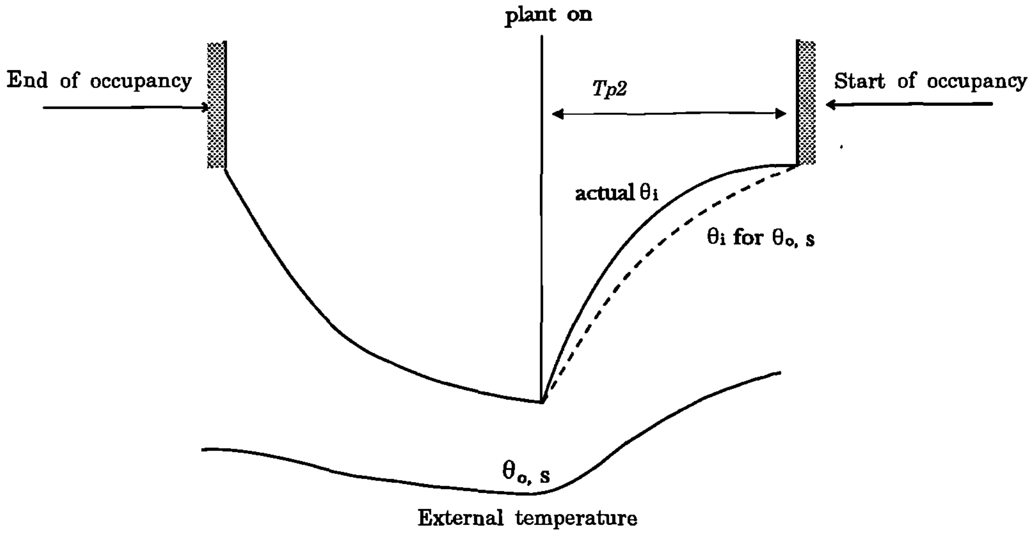


Fig.4.12 Effect of varying θ_0 during the pre-heat time

(after McLaughlin, ref.82)

A change in θ_0 during the pre-heat time means less thermal loss through the walls and thus less energy is required to reach the set temperature. The rule shown below is included to check if the error is due to a change in external temperature,

```
if   ZoneX.mode_status is 'start'
     & ZoneX.temp_in_measured ≥  ZoneX.temp_set
     & ABS[ZoneX.optimum_start_actual - ZoneX.optimum_start_predicted] > 300
     & ABS[ZoneX.temp_out_measured - ZoneX.temp_out_predicted] > 1
     & time_measured between 0600 and 1000
⇒    Ch_in_ext_temp is confirmed
     & [0.5× (ZoneX.temp_out_measured / ZoneX.tenp_out_predicted+1)]
       is assigned to ZoneX.ext_temp_factor_predicted
```

The condition clauses in this rule check to see that the controller is in 'start' mode and that its set point has been reached. This rule has a higher inference priority than the rules that allow the controller to leave the 'start' mode; this ensures that those rules are evoked following this rule. If the pre-heat time is less than predicted, and the current value of the external temperature is greater than at the start of the pre-heat period, the value of the external temperature factor is changed. This value is another property belonging to the 'Building' class and is used to modify the value of the external temperature passed to the optimum start time function to take account of a rise in the external temperature. It is assumed that this will only happen during the morning start up period between the hours of 6am and 10am. The optimum start rule uses the result of *ZoneX.temp_out_measured* × *ZoneX.ext_temp_factor_predicted* in the thermal model to predict the pre-heat time. This value is also stored in the *ZoneX.temp_out_predicted* property before the controller enters the start mode. A corresponding rule exists to decrease the external temperature factor when the pre-heat time is too short.

For example, if on a particular day of the year the external temperature changes from 2 degrees to 5 degrees during the pre-heat time, the external temperature factor becomes 1.75 and thus 3.5 degrees is passed to the function as the external temperature. This is taken as the average value during the pre-heat time. It is assumed that the external temperature change is linear and that the same change occurs each day for a particular time of the year, in both cases this will only be an approximation. A more accurate solution would be to monitor and predict external temperatures in a similar way to the occupancy data.

If no significant change has occurred in the external temperature the error may be due to a change in thermal constants of the room or the heating system. This may be due to the room having recently been partitioned or furnished differently and is less likely than the previous cause. The following rule is used to detect an error in the thermal characteristics.

```
if  ZoneX.mode_status is 'start'
    & ZoneX.temp_in_measured > ZoneX.temp_set
    & ABS [ZoneX.optimum_start_actual - ZoneX.optimum_start_predicted] > 300
    & ch_in_ext_temp is False
⇒   Ch_in_thermal_constants is confirmed
    & calculate new ZoneX.a_predicted
    & calculate new ZoneX.b_predicted
```

If an error in the pre-heat time greater than 5 minutes occurs a rule is triggered which corrects the model. The rule backward chains to the *ch_in_ext_temp* rule to check that the error was not due to a change in the external temperature. The model is modified by changing the constants to reduce the error. Data is stored for the pre-heat and cooling cycles and this is used to find the correct thermal constants using re-arrangements of the optimum start and stop functions.

Equation (5) can be represented as,

$$t = -a \ln\left(\frac{\vartheta - b - \theta_e}{\theta_o - b - \theta_e}\right) \qquad ..... \qquad (7)$$

$$\text{where}, \; a = \frac{M.S_a + m.S_s}{A.U_f} \; \text{and} \; b = \frac{Q_e}{A.U_f}$$

The value of a represents the room's thermal time constant and b the theoretical maximum temperature differential above the external temperature which the room can ever reach with the heating on. The values a and b are stored in the building data-base for each thermal time constant. In the prototype two sets were used, one for shut down times less than 12 hours and one for more than 12 hours, i.e. a weekend. Further values stored are,

t_1 = Time to heat from θ_{01} degrees to ϑ_1 degrees [s].
t_2 = Time to cool from θ_{02} degrees to ϑ_2 degrees [s].
ϑ_1 = Minimum room temperature at the start of occupancy [°C].
ϑ_2 = Minimum room temperature at the end of occupancy [°C].
θ_{01} = Initial room temperature at the optimum start time [°C].
θ_{02} = Room temperature at the optimum stop time [°C].
θ_{e1} = External temperature during optimum start [°C].
θ_{e2} = External temperature during optimum stop [°C].

Two equations can thus be formed using expressions 5 and 6 which include a and b.

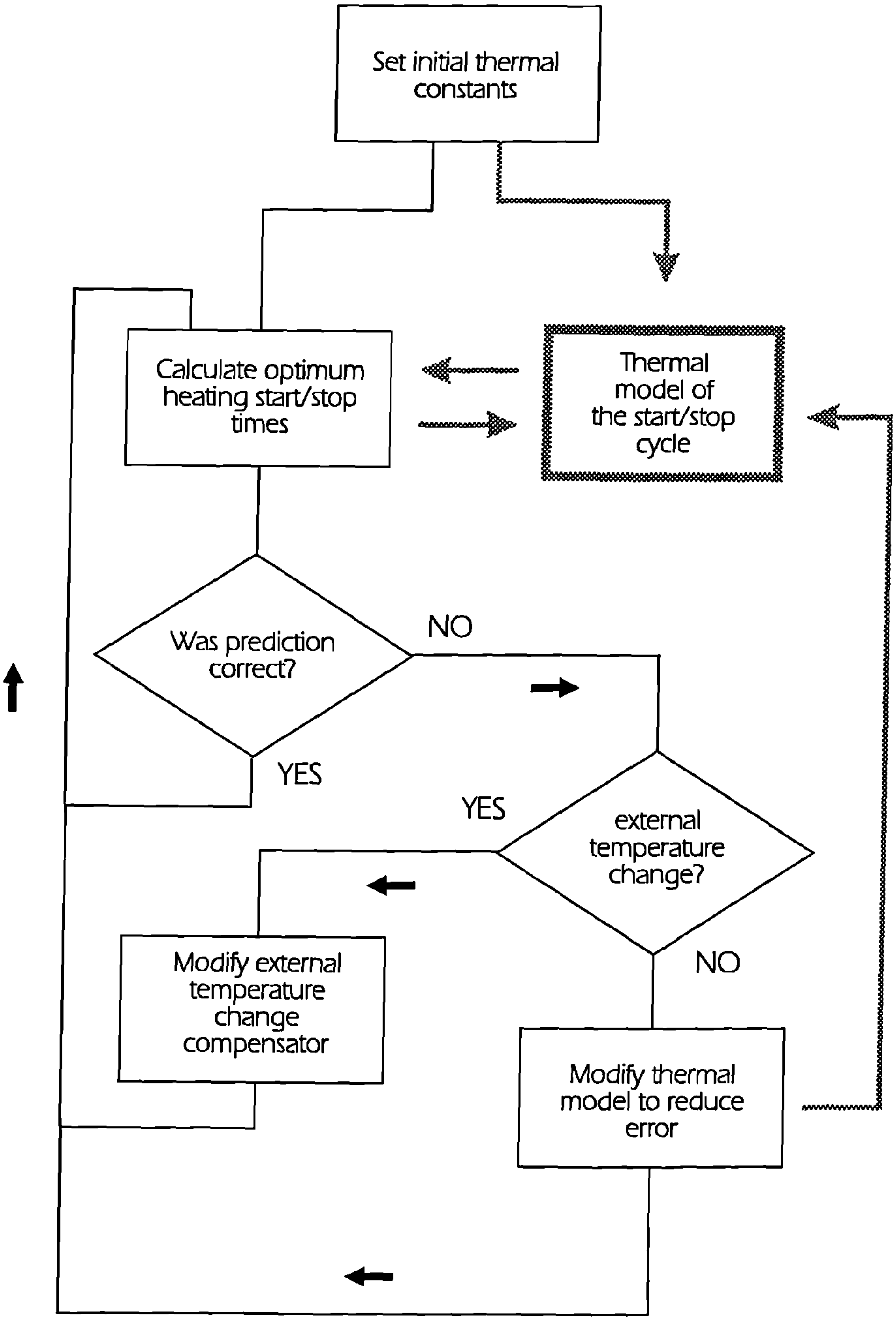


Fig.4.13 Block Diagram of Zone Heating Control

$$t_1 = -a \ln\left(\frac{\vartheta_1 - b - \theta_{e1}}{\theta_{o1} - b - \theta_{e1}}\right) \quad \ldots\ldots \quad (8)$$

$$t_2 = -a. \ln\left(\frac{\vartheta_2 - \theta_{e2}}{\theta_{o2} - \theta_{e2}}\right) \quad \ldots\ldots \quad (9)$$

rearranging 9 in terms of a gives,

$$a = -t_2 \Big/ \ln\left(\frac{\vartheta_2 - \theta_{e2}}{\theta_{o2} - \theta_{e2}}\right) \quad \ldots\ldots \quad (10)$$

and b is found by re-arranging 8 and substituting into it the value of a found by equation 10,

$$\Rightarrow e^{-t_{o1}/a} = \frac{\vartheta_1 - b - \theta_{e1}}{\theta_{o1} - b - \theta_{e1}}$$

$$b = \left(\frac{\theta_{o1} . e^{-t_1/a} - \vartheta_1}{e^{-t_1/a} - 1}\right) - \theta_{e1} \quad \ldots\ldots \quad (11)$$

This self-tuning model is illustrated in Fig.4.13 which summarises the zone heating control. At the commissioning stage it is decided whether or not the area is light weight or heavy weight and initial values for the thermal constants are chosen. The accuracy of these is not important as they will adapt with time to optimum values for the zone under control. The occupancy predictor calculates the time when heating is required by interpolation of the global occupancy data-base. The optimum pre-heat time is calculated using equation (5) so that the required environmental conditions are reached at the start of occupancy. The thermal constants used in equation (5) are selected based on the length of the cooling time for the area. When the desired temperature is reached, rules check to see if the preheat time was correct. If there is a significant error the rules first check to see if the external temperature changed during the pre-heat time. If the external temperature was as predicted, the rules make a change to the thermal constant values using an iterative procedure. The thermal model will thus adapt to changes within both the heating system and the room. Results given in Appendix D show that the thermal model was found to predict optimal start and stop times to an accuracy of around 3% for various external temperatures.

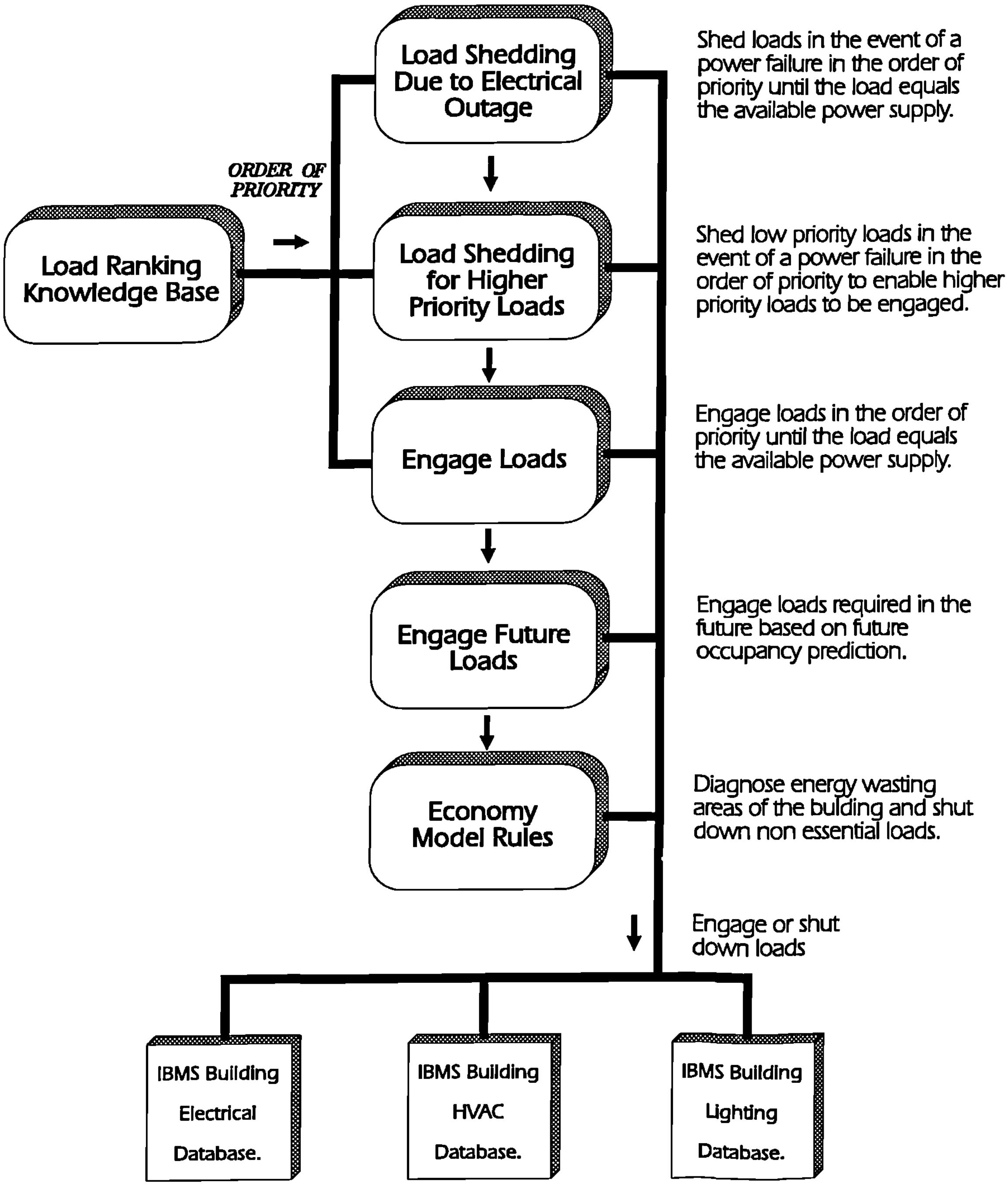


Fig.4.14 Energy Rule-Base Structure Overview

4.3 Global Controller

The global control knowledge-base, which is run on the central IBMS controller, is structured into separate rule-bases which are suggested in an order of priority [Fig.4.14]. These rules provide global energy savings and emergency handling features. The highest priority rules are suggested if load shedding is required and loads are shut down in the order of load priority until the buildings electrical load equals the supply capability. The order of priority for loads is a dynamic variable which is determined by an area's function, occupancy and the time of day. If no further load shedding rules can be suggested the 'engage loads' rules are suggested; loads are then engaged, if required, providing the supply capability is sufficient. The final group of rules are included to detect energy wasting areas within the building such as dormant rooms and to shut down unnecessary loads to save running costs.

These rules must be written such that they are independent of the number of the outstations. They operate on the 'Building' class and make extensive use of pattern matching and strategic resetting of the rule hypothesis.

4.3.1 Global Control Override Rules

The rules described in this section have hypothesis objects which are automatically reset and evoked at regular intervals. The implementation is described in Appendix C. Through the use of class pattern matching they operate on all 'Building' and 'Plant' objects irrespective of the number of objects in each class.

4.3.1.1 Related area control

There are many areas which are not defined as individual zones such as corridors and wash rooms. These areas may not be continuously occupied and thus a self-tuning zone controller would have difficulty in predicting occupancy along with the pre-heat and optimum stop times. The control of these areas is therefore linked to the activity of associated areas within the building thus ensuring that these areas are always at the required environmental conditions should they become occupied[25]. Thus a wash room will be defined as 'quiet' even though it is not occupied if there is occupancy in the rooms near to it. The room is controlled by zone controllers as described previously but without the self-learning occupancy prediction feature; the occupancy prediction is down loaded to the controller from the global

---

[25] Relaxed set points may wish to be assigned to such areas.

control system[26]. For example, the following rule sets the control mode for areas of the building defined as 'corridor'.

```
if   |BUILDING|.description_set = 'corridor'
  &  |BUILDING|.mode_status = 'off'
  &  ||BUILDING||.mode_status is not ('on', 'start')
  &  |BUILDING|.floor_set = ||BUILDING||.floor_set
  &  MAX[|BUILDING|.no_set]
⇒    Related_area is confirmed
  &  'start' is assigned to |BUILDING|.mode_status
  &  reset Related_area
  &  do Related_area
```

The first pattern matching finds the set of objects belonging to the class 'Building' that are defined as "corridors" and whose control mode is 'off'. The second independent pattern matching, indicated by the double lines, finds the set of objects whose mode is either 'on' or 'start'. The first set is further reduced to those objects which have the same *floor* property value as any of those objects in the second set. These are the objects representing rooms in the building that are in the 'off' control mode whilst another area on the same floor is either in pre-heat or occupied mode. These areas are switched to 'start' mode. This has the effect of keeping them in perpetual pre-heat until they become occupied then their controller will switch to the 'on' state. A maximum function is used to select just one object from the set and as the actions of the rule reset the rule's own hypothesis the rule is continuously fired until all the areas meeting the condition of the existential pattern matching are switched to 'start'. The rule will then fail as by definition of the rule's function no objects with the same value *floor_set* property will simultaneously meet the two pattern matching conditions. A similar rule switches areas from 'on' to 'stop' if all the areas on the same floor enter the 'stop' or 'off' states.

4.3.1.2 Integration with the planned events database

As mentioned previously, a provision must be included to override the zone predicted occupancy when an area is required to be used at a time of the day when it is not usually occupied. If this is known in advance, the IBMS operator creates a record in the planned events database listing the location, time and duration. This creates an object in the class 'Diary' which is used by the global control knowledge-base for overriding the settings of the zone controllers. 'Diary' is a sub-class of the parent class 'Planned_events' and has properties of *start_time_set*, *stop_time_set*, *event_no_set* and *reason_set*. The following rule, which makes use of the computer's real time clock, searches the planned events database and switches appropriate controllers to the 'start' mode.

---

[26] This does not effect the control of lighting which remains the same as for a standard zone controller.

```
if   ||Diary||.reason_set = 'occupied'
     & |BUILDING|.location_no_set = ||Diary||.location_no_set
     & |BUILDING|.optimum_start_predicted + time_measured > |Diary|.start_time_set
⇒    Override_start is confirmed
     & 'start' is assigned to |BUILDING|.mode_status
```

Pattern matching is used to find areas of the building that have an occupancy override record in the Diary database. If the start of planned occupancy occurs at a time in the future which is less than or equal to the predicted optimum starting time for the location in question, the hypothesis *Override_start* is confirmed. The rule actions set the operating mode for all these locations to 'start'; this is then downloaded to the controllers via the LAN. The zone controller will not immediately switch itself back to 'dormant' using its own predictor as under the 'start' control rule conditions the controller must remain in the 'start' mode until the set points are reached. Only then may it switch back to 'off' if occupancy did not occur. In this way, the zone controller's activity rules are overridden without disabling them.

Another rule exists to switch areas to 'stop' mode when the *stop_time_set* property in the 'Diary' class object equates with the optimum stopping time for an individual area. Similarly, plant equipment can be shut down automatically for preventative maintenance. This makes use of the 'maintenance' sub-class of the 'Planned events' class. Each object of type 'Maintenance' has properties *plant_no* and *class* along with the time properties inherited from its parent class. When equipment is shut down for maintenance the control status is set to -2 which signifies that either preventative maintenance is being conducted or that a breakdown has occurred. For example, the following rule shuts down the air handling units in areas where maintenance is scheduled to change the filters.

```
if   |PLANT|.control_status > 0
     & |PLANT|.class_set = 'AHU'
     & ||Maintenance||.class_set = 'AHU'
     & ||Maintenance||.code_set = 38
     & |PLANT|.no_set = ||Maintenance||.no_set
⇒    Ahu_Maintenance is confirmed
     & -2 is assigned to |PLANT|.control_status
```

Pattern matching is used to find the set of plant objects which represent air handling units which are either fully switched on or are under closed loop control. A second independent pattern matching, indicated by the double lines, finds the set of 'Maintenance' class objects which represent planned filter replacement; this work is coded in the fault database as fault code 38. The set of plant equipment objects is further reduced to objects which share the same air handling unit number as those in the 'Maintenance' class sub-set. The control level for the plant equipment objects remaining in this sub-set is set to level -2 by the rule's action. Similar rules exist for different classes of plant equipment and fault codes.

4.3.2 Global Energy Saving Rules

The study includes the option of running the BMS in an 'economy' mode, whereby the rule-based global control system can further reduce energy consumption within the building. Some of these features, such as load control and load balancing, are incorporated in existing energy management systems. The aim here is to show that the existing algorithms can be assimilated in a rule-based methodology. These features can be implemented in such a way that they cause less discomfort to the majority of the occupants whilst still saving energy.

4.3.2.1 Economy mode

At certain times of the day HVAC and lighting can be reduced in sparsely occupied areas of the building to save energy. In areas which are defined as 'quiet', the HVAC set points are reduced before the optimum stop time thus saving energy and causing a slight discomfort to only a small number of people. This is achieved by the dormant mode control rules which take the form shown below.

```
if   BUILDING.mode_status is 'Economy'
     & |BUILDING|.activity_actual is 'quiet'
⇒    Economy_control is confirmed
     & Building.temp_min_set is assigned to |BUILDING|.temp_set
     & Building.light_min_set is assigned to |BUILDING|.light_set
```

This rule makes use of the *Building.mode_status* property, the value of which is set by the operator as being 'normal' or 'economy'. This property is used to enable or disable the economy rules. The property *Building.temp_min_set* stores the minimum set values for the economy mode. This is down loaded to all zone controllers which have a 'quiet' activity level. If the economy mode ends, or if the activity in these areas change, the following two rules down-load the zone's normal set points.

```
if   Building.mode_status is 'Normal'
⇒    Normal_control is confirmed
     & |BUILDING|.temp_normal_set is assigned to |BUILDING|.temp_set
     & |BUILDING|.light_normal_set is assigned to |BUILDING|.light_set
```

```
if   Building.mode_status is 'Economy'
     & |BUILDING|.activity_actual is not 'quiet'
⇒    Normal_control is confirmed
     & |BUILDING|.temp_normal_set is assigned to |BUILDING|.temp_set
     & |BUILDING|.light_normal_set is assigned to |BUILDING|.light_set
```

These rules are reset with changes in global data at regular time intervals as outlined in appendix C.

4.3.2.2 Load Control

Other forms of energy savings incorporated in the global control rules are load control, load balancing and peak lopping. The load control technique relies on the fact that the heating, ventilation and air conditioning plant is sized for design conditions which may only apply for a few days each year. For the remainder of the year the plant is running below its design capacity. Although conventional temperature controls may be used to regulate the amount of heating or cooling energy coming into the building, they do not necessarily do anything to limit the amount of air being circulated via a fan driven ventilation system. As long as the fan is running, electrical energy is being consumed just in moving the air around the building. To save electrical energy, load cycling switches the plant off for a percentage of the time. Switching a fan off for 10 minutes in every hour saves 16.7% of the electrical energy driving the fan[27]. This saving is made at the expense of the occupant's comfort but, providing the limits to which load control operates are carefully chosen, the effect should be virtually unnoticeable to the occupants.

The air handling units selected for load control supply areas of the building where the actual environmental conditions are near to their maximum set temperature values. This differs from load shedding in that loads are being alternated rather than shut down. This switching must be updated much faster than the re-allocation of load priorities and subsequent changes described later in the section on load shedding. Also, unlike load shedding, the autonomous control operation of the zone outstations is only overridden and not completely suspended. The following describes the set of rules which handle load control for the air handling unit fans.

```
If    Building.mode_status = 'Economy'
  & |BUILDING|.mode_status = 'on'
  & |BUILDING|.temp_measured ≥  |BUILDING|.temp_set ×  0.9
  & ||HVAC||.location_influence_no_set = |BUILDING|.location_no_set
  & ||HVAC||.class_set = 'Air handling units'
  & ||HVAC||.time_off_actual ≤  ||HVAC||.max_time_off_predicted
⇒   Ahu_load_control is confirmed
  & 'stop' is assigned to |BUILDING|.mode_status
  & 0 is assigned to ||HVAC||.control_status
```

---

[27] This does not necessarily save any of the heating or cooling energy required to meet the set environmental conditions. If by switching the fan off for 10 minutes the zone temperature changes by 1°C, additional heating or cooling energy is then needed during the 'on' period to get it back to the required temperature. This extra energy would be roughly equal to the heating or cooling energy saved.

In the load control rule shown on the previous page, pattern matching is used to find the areas in the building where the temperature is approximately equal to the maximum pre-set limit. The sub-set of HVAC objects are found which serve these areas and the set is further reduced to those which are air handling units which are currently running. The status of these areas is set to 'stop' which is used to signify a temporary shut down. When the heating level drops to the minimum set point, the heating can be re-engaged under zone control (in the same way that an optimum stop cycle is aborted if the occupancy is still greater than one when the pre-set minimum temperature is reached). Therefore, a further rule is not required in the global control knowledge-base to re-engage the loads. The rule also checks that the air conditioning is sufficient for the number of occupants in the area using a method to calculate the optimum value of the *max_time_off_predicted.* This calculation is based on the current EEC guidelines for ventilation requirements in buildings which are summarised in Appendix C. Alternatively CO_2 monitoring could be used but this would incur the extra cost of installing CO_2 sensors.

If N_{ac} is the minimum number of air changes required per hour, Q_f is the fan's rated air supply and V is the room's volume, the property *max_time_off_predicted* (in minutes) is calculated as follows:

$$N_{ac} = Q_f \times 60 \times 60 / V \times (60 - max\_time\_off\_predicted)/60$$

$$\Rightarrow max\_time\_off\_predicted = 60 - 60.N_{ac} /(Q_f \times 60 \times 60 / V)$$

The required number of air changes is calculated by,

$$N_{ac} = 0.001 \times Q_c / H$$

where H is the room's height and Q_c is the required air flow in $ls^{-1}m^2$ (floor area) for comfort calculated by:

$$Q_c = 10.\frac{G}{C_i - C_o}.\frac{1}{\varepsilon_v} \qquad ..... \qquad (12)$$

where,

Q_c = ventilation rate required for comfort (ls^{-1})
G = sensory pollution load [olf]
C_i = perceived indoor air quality, desired [decipol]
C_o = perceived outdoor air quality at air intake [decipol]
ε_v = ventilation effectiveness

For example, the main lecture theatre at Brunel has a floor area of $315m^2$, a volume of $1150m^3$ and a maximum seating capacity of 200 people. The ventilation fans for supply and extraction are rated at 4.3A 3ϕ and supply air at $4m^3/s$. If the theatre is only occupied by 70 people, the required ventilation calculated using (12) would be 1.01 l/s. This means that the theatre would require one air change per hour and the fan's duty cycle would be 0.08. This would save 5.7kW of electricity per hour.

Load control does cause additional wear on the plant because of the increased starter operations, increased strain on the fan motors and fan belt wear. While the latter is true, most manufacturers of air handling units rate their products for much greater frequency of operation than is imposed by load control. The energy savings offset the costs of additional belt replacement many times over. However, to avoid extra maintenance, two speed fans can be employed and the fans switched from high speed to low speed thus reducing the adverse effect of the sudden reduction in air movement.

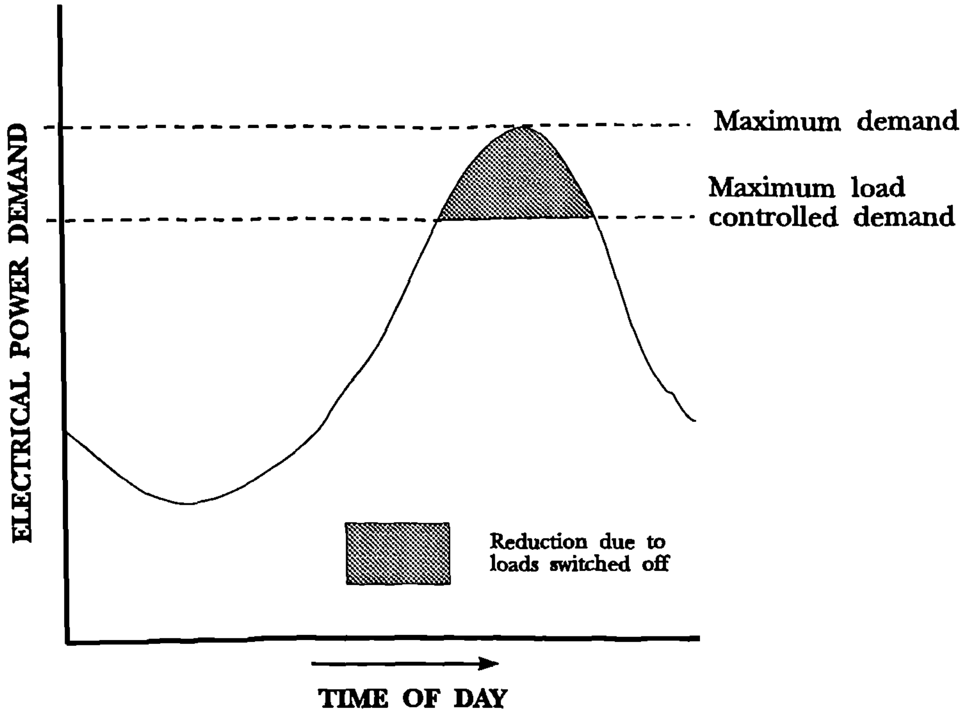


Fig.4.15 Diagram Showing Power Demand Control
(after Scheepers, ref.10)

Load balancing can be integrated with the load control rules. This involves the temporary and systematic switching off of different electricity consumers to reduce the peak electricity demand. Large electricity consumers are charged a tariff based on the peak demand. This is because the installation must be of sufficient capacity to meet this maximum demand and hence it mostly reflects the overheads in terms of the supply installation. Peak lopping is designed to reduce this peak as far as possible. Fig.4.15 shows a typical demand for a building during the day. A peak often occurs mid morning or mid afternoon. If this peak, which may only last for a few minutes, is reduced the fixed charge tariff would also be reduced.

In conventional systems the loads are assigned a weighting in terms of their size and the load balancing algorithm arranges the start and stop periods for each load so that taken over the maximum cycle period the total load at any one time is as close to the average as possible. One cannot assume that all the loads are the same or that the thermal time constants of the areas they control are similar. For example, at worst case the loads may consist of one 100kW load and twenty 4kW loads. Similarly, there may be one large area with a pre-heat time of 1½ hours and others which require only 20 minutes. Within the rule-based control system the load balancing algorithm can be incorporated by the addition of a single rule. This rule is a modification of the load control rule shown previously. Where indicated by the dotted lines the rule is exactly the same as the load control rule but incorporating five extra condition clauses and two extra action clauses. Also, the pattern matching reduces the main sub-set to one object which is shut down by the rule actions unlike the load control rule which can operate on a set of eligible objects simultaneously. The rule resets its own hypothesis and calls itself if the condition clauses are all true. Thus the hypothesis is proved false when all the eligible loads are shut down. The rule is shown below.

```
If   .................
  & |||BUILDING|||.mode_status = 'stop', 'on'
  & ||HVAC||.location_influence_no_set = |||BUILDING|||.location_no_set
  & ||||HVAC||||.class_set = 'Air handling units'
  & SUM[||HVAC||.power_predicted]
        ≥  ½ ×  SUM[||||HVAC||||.power_predicted]
  & MIN {  ABS {SUM[||HVAC||.power_predicted]
      - ½ ×  SUM[||||HVAC||||.power_predicted] - ||HVAC||.power_predicted} }
⇒  Ahu_load_balance is confirmed
  &  .................
  & reset Ahu_load_balance
  & do Ahu_load_balance
```

The first three additional clauses of the rule find the set of air handling units which control areas which are either in the 'on' or 'stop' modes. This set of 'Hvac' equipment is indicated by the four lines. The fifth additional clause checks that the sum of the power for all the engaged air units is greater or equal to half of the total power consumption for all the air handling units not supplying a dormant area. This prevents more fans being shut down, even thought is economic to do so, if more that half the maximum theoretical demand is temporarily shut down. The addition of this clause controls the peak demand such that it cycles about the mean value and prevents excessive peaks as illustrated in Fig.4.16.

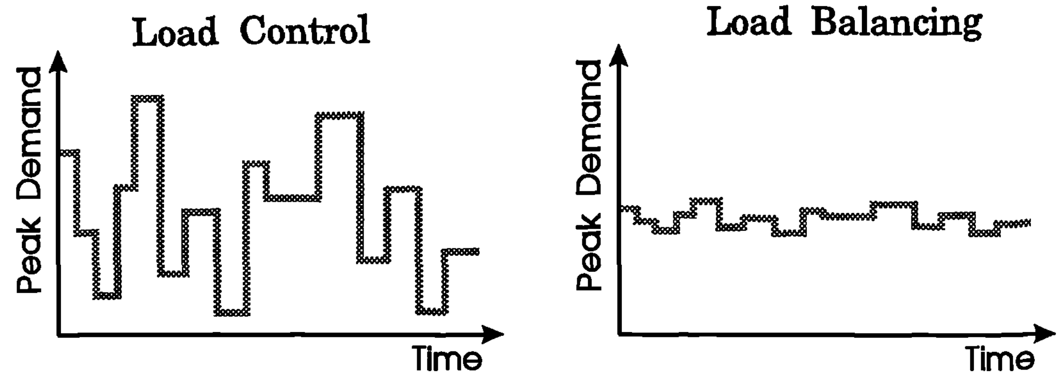


Fig.4.16 Peak Demand Profile for Balanced/Unbalanced Load Control

The final additional clause reduces the set of eligible loads to be shut down, indicated by the double lines, to the one that brings the total demand to as close as possible the mean value. This is done by finding the single object that gives the minimum absolute value when its power rating is deducted from half the total theoretical power minus the engaged power for the air handling units.

4.3.2.3 Sundry Energy Gains

The zone controller could take into account additional heating gains or losses other than the heating or cooling systems. Sundry gains to buildings can comprise a high proportion of the input energy. In lecture theatres, auditoriums, leisure theatres and other public meeting places, the sensible and latent heat gains from people may dominate the heat balance equation. The actual heat gain from people depends upon the rates of working, the internal environmental dry-bulb temperature and the relative humidity of the air. Table 4.2 gives some approximate figures for the rate of heat production for different activity levels [82].

Activity	Wm^{-2}
Sleeping / seated quietly	40-50
Office work	60-80
Walking at 3mph	150
Machine work	100-260
Squash / very heavy work	290-420

Table 4.2 : Rate of Heat Production for different activity Levels
(From 'Energy Management' by O'Callaghan, ref.83)

There can often be heat gains from electrical devices in the building. Electricity for lighting can constitute a major heat gain. For example, an 80W fluorescent fitting needs 100W of power supplied and the extra 20W is liberated directly from the control circuitry as heat. The same is true for electrical motors.

The closed loop control system will compensate for these gains during the occupied mode of operation. This information could be used to modify the thermal model for optimum start by altering the value of the assumed energy input. However, by definition at the optimum start time, the occupancy and hence the lighting load will be zero. At the start of the optimum stop cycle this is not the case and knowledge of this extra heat gain can be used to bring the optimum stop time forward and hence save energy. The IBMS 'PLANT' class stores information on plant activity, power rating and their physical location. This information can be used to predict the sundry heat gains for each object in the 'BUILDING' class. If these gains are significant, Q is no longer taken as zero in the optimum stop model. The following rule updates the sundry gain properties of all 'BUILDING' objects to take account of occupancy.

```
if   |BUILDING|.occupancy_measured > 0
⇒   |BUILDING|.occupancy_meaured × 80
        is assigned to |BUILDING|.Q_sundry_predicted
```

Moreover, knowledge of sundry gains can be used by the global control system to make decisions on starting large heating or cooling plant. It is common in large buildings to have more than one size of boiler. When the demand is low it is more efficient to use a smaller boiler and vice versa. Knowledge of the building's occupancy, the predicted sundry gains and the mathematical thermal model can be used to start the minimum amount of plant required to meet the predicted demand. This will provide further energy saving.

4.3.3 Emergency Situation Handling Rules

This section outlines how the rules running on the global control knowledge-base can be used for handling emergency situations. One such example of this is the loss of electrical power but others might include emergency evacuation in the event of a fire or a security threat.

4.3.3.1 Priority Ranking Knowledge-Base

All electrical loads are ranked in order of priority. This rank is used to determine the order of load shedding and load re-engagement during a supply outage [Fig.4.17]. The rank also governs the priority of loads under load control. The ranking order is not specific to a load; i.e. more than one load can have the same ranking value. Each object in the 'PLANT' class has a *rank_actual* property which is of type integer and can assume a value between 0 and 1000; the lower the number the more important the load.

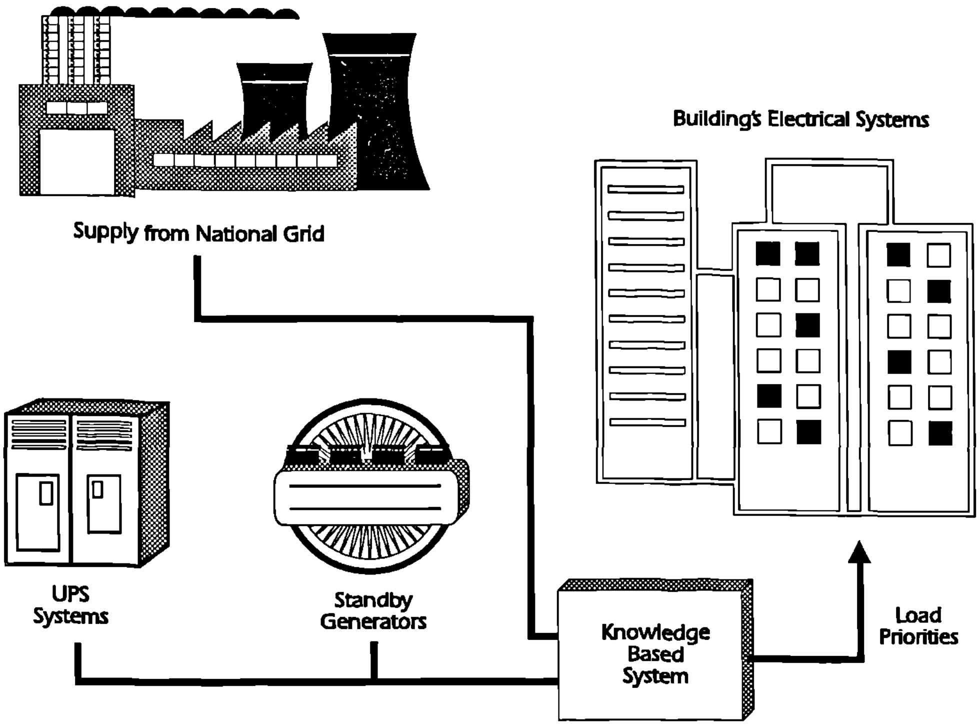


Fig.4.17 Knowledge-based Ranking of Loads

Each sub-class of plant equipment inherits initial property values at run time which are inherited to all their objects. Thus all the equipment of the same class initially has the same rank number. The rules in the rank rule-base modify these initial priority settings based on the activity within the building. For example, the light level in an area is given a priority rating based on the area's occupancy and usage factors. In general, under load shedding conditions, the lowest priority is given to unoccupied areas. All non essential electrical loads are given lowest priority followed by HVAC loads. Priority is next given to non important electrical loads, followed by lighting then important loads. The last 20% of lighting and the essential loads are given the highest priority. The building should have enough power generation capacity or UPS systems to keep essential loads such as computers and emergency lighting running.

The values of the rank properties are free to change during the processing and a separate rule-base is used to determine the ranking order. The value of a load's *rank_actual* property is determined by rules based on the following knowledge:

- The activity in the area served by the load.
- The function of a particular area served by the load.

- The time of day.
- The status of related plant equipment.
- The day of the week or season.
- The importance of the load to the function of the building (irrespective of occupancy).
- The cost of stopping and restarting plant.
- The related area's predicted occupancy.

The property *reason_actual* is of type string and stores an explanation in plain language regarding the ranking decision for a particular load. This data is later used in the message log to describe the system behaviour to the operator. The message log and the operator's control screen are both described in the next chapter. A set of rules prioritises loads for each class of plant equipment. Each set has the same hypothesis object, the name of which takes the form 'Plant-type_ranking', e.g. rules to rank pumps all have the hypothesis *Pump_ranking*. This is used by global load shedding and engage rules to strategically reset all the ranking rules for a sub-class of plant equipment. This makes the rule processing more efficient and saves time when backward chaining as only rules effected by the changes in data are reset.

Each electrical load has an *importance_set* property which can be pre-defined by the IBMS operator. This can take on the value 'low', 'medium' or 'high'. This enables the rules to rank loads depending on the BMS operator's perceived importance. For example, the chairman's office can be defined as 'high' whilst a store room may be defined as 'low'. In general, if not pre-defined, most areas will take on the default value of 'medium'. The rules then rank loads depending on their relative importance and area occupancy factors. For example, an occupied store room has priority over a dormant office even though it may be of less importance than the office.

An example of a lighting ranking rule is shown below.

```
If   |Lighting|.light_level_set > 80%
  & |Lighting|.importance_set = 'high'
  & ||BUILDING||.activity_actual is 'occupied','active'
  & |Lighting|.location_no_set = ||BUILDING||.location_no_set
⇒   Lighting_ranking is confirmed
  & 250 is assigned to |lighting|.rank_actual
  & "Full lighting is a low priority even in occupied areas" is assigned to
      |Lighting|.reason_actual
```

Under load shedding conditions, lighting is incrementally reduced in steps of 20% from the initial set value until all the lighting is switched off. The lower the light level the higher its priority is set. The above rule sets the priority for a 20% reduction in lighting for all lighting loads which have been pre-defined as 'high' importance and which are currently set to above 80%. The ranking order is used for re-engaging loads as well as load shedding. Therefore it does not matter if the load is currently switched on or off as account is taken for this in the *Load_shedding* and *Load_engage* rules. Pattern matching is used to detect the required location object from the class '*Building*' thus allowing it to be a dynamic object.

In total there are 36 rules for ranking lighting. Rules exist for current light levels above 80%, 60%, 40% and 20%. For each of these there are rules for differing pre-defined importance and current activity. This assures that minimal lighting in occupied areas is given highest priority and lighting in unoccupied areas is given lowest priority. A list of ranking priorities is given in Appendix C. Ranks above 900 are considered non important loads which have no economic basis to be engaged, such as the heating and lighting in dormant areas. Loads with ranks under 400 are defined as very important and those under 300 are deemed to be essential.

For the case of the building's electrical systems, the priority depends on the occupancy in the area, the area's function and the time of the day. At lunch time priority can be given to the kitchens and dining rooms and the lighting and HVAC levels can be relaxed in the office and production areas. For example,

```
if   |BUILDING|.description_set is 'kitchen'
  & Week_day is True
  & Lunch_time is True
  & |PLANT|.location_no_set = |BUILDING|.location_no_set
=>  Electrical_ranking is confirmed
  & |PLANT|.rank_actual + 5 is assigned to |PLANT|.rank_actual
```

In the mid afternoon, the situation may be reversed and if the chairman's office is occupied a higher priority is given to that area. The priorities of pumps change throughout the day; the cold drinking water has more priority during office hours on weekdays. Such rules require knowledge of the time of day and week. This is achieved using rules which define certain times and dates using the computer's real time clock. These rules have hypothesis such as *Week_day*, *Office_hours*, *Summer* and *Daylight_hours* and are all linked to the class '*GLOBAL*'. These are proved true if the corresponding time and date conditions are true. These hypothesis objects are then used in the conditions of the ranking order rules. Linking these to the class '*GLOBAL*' allows them all to be reset when any of the conditions change. This implementation is explained in Appendix C.

The ranks of various loads can be modified based on secondary considerations. For example, if an area is currently dormant its services will be given a low priority. However, if the area is predicted to be becoming occupied the rank of services serving this area is increased by one to set them above other zones in this priority band. Similarly, if the temperature is at or near the maximum set point the priority is reduced by one. When load shedding only occurs for a short time minimal discomfort will occur as the area will take time to cool below the minimum set point. This is assuming that the outage is temporary, after a sustained outage and a forced shut down of heating or cooling, these rules will not be evoked.

```
if    |BUILDING|.activity_actual is not equal 'Dormant'
  & |BUILDING|.temp_measured >  |BUILDING|.temp_set - 1
  & ||HVAC||.class_set = 'Air handling unit'
  & ||HVAC||.location_influence_no_set = |BUILDING|.location_no_set
⇒   Ahu_ranking is confirmed
  & ||HVAC||.rank_actual = ||HVAC||.rank_actual -1
```

The above rule example uses pattern matching to select all the building class objects that are not dormant and those which are near to their set point. A window of one degree temperature variation from the set value is used. A second existential pattern matching finds the sub-set of the 'Hvac' class of air handling unit objects and then the further sub-set which matches the *'BUILDING'* class sub-set for locations of influence. The rank is then reduced by one for all objects in this sub-set. These rules have a lower inference priority to ensure that they are evoked after their associated location based ranking rules.

4.3.3.2 Load shedding rule-base

When the power is lost from the main incoming feeder, the building management system will start the emergency power supply and invoke its load shedding procedure. When a load shedding situation is detected, the string 'out' is written to the *mode* property for all objects belonging to the class 'Building'. Once downloaded to the outstations this has the effect of placing the zone controllers into a state which they have no means of leaving thus suspending their autonomous control. Zone control is then temporarily passed to the global control system. The load shedding rules then fire until the power equilibrium is reached, i.e. the power demand equals the power supply capacity. A load shedding rule exists for each sub-class belonging to the parent class 'PLANT'. This need not be an individual item of plant as blocks of lighting which are switched together are defined as single objects. By use of an iterative rule-based algorithm, loads are incrementally shed depending on the load's priority. The shedding rule for the 'lighting' class is shown overleaf.

```
If    SUM[|PLANT|.power_measured] + Electrical.power_required_actual ≥
          Electrical.supply_capacity_set
   & is |PLANT|.control_status ≠ 0,-2
   & is MAX [|PLANT|.rank_actual]
   & |PLANT|.class_set = 'lighting'
   & |PLANT|.light_level_set > 20
⇒    Load_shed is confirmed
      & |PLANT|.light_level_set - 20 is assigned to |PLANT|.light_level_set
   & -1 is assigned to |PLANT|.control_status
   & reset Lighting_rank
   & do Lighting_rank
   & reset Load_shed
```

The first condition clause checks to see that more load shedding is required. Pattern matching is then used to find the set of objects belonging to the 'PLANT' class which are not switched off. The control status value of -1 is used to signify that this load's closed loop control set point has been reduced by the load shedding rules. In the case of lighting, this is an incremental reduction in the percentage light level, for heating it is a temperature set point reduction. The set point is then reduced for the object with the highest value in its *rank* property. If this is a lighting object and its set level is above 20%, the set level is reduced by 20%. A similar rule exists to set the status to 0 if the set light level is currently at 20% or less. The *Lighting_rank* hypothesis is reset which forces the lighting ranking rules back onto the agenda, this load is then given a lower rank for the next time this rule is fired. Finally, the object *Load_shed* is reset which forces re-evaluation of all the load shedding rules. This continues until the power equilibrium is reached or all the loads are switched off. In either case, the hypothesis *Load_shed* will eventually become false.

```
If    SUM[|PLANT|.power_measured] + Electrical.power_required_actual ≥
      Electrical.supply_capacity_set
   & is |PLANT|.control_status ≠ 0
   & is MAX [|PLANT|.rank_actual]
   & |PLANT|.class_set = 'pump'
⇒    Load_shed is confirmed
   & 0 is assigned to |PLANT|.control_status
   & reset Pump_ranking, reset Load_shed
   & Do Load_shed
```

Similarly, for the class of pumps the loads are switched off by the above rule.

4.3.3.3 Load Shedding for higher priority loads

If a system of dynamic priorities is used a feature should be provided to change priorities under load shedding conditions. For example, at the start of load shedding the executive suite may have been occupied and its heating and lighting was given a high priority. This meeting may now have ended so loads which were shed previously in other parts of the building can now be re-engaged. A 50kW load which was turned off may now have a greater priority than ten other 5kW loads which were left running. The rule-base needs to identify this change, update the priorities and shut down lower priority loads in order to re-engage this new load. This situation is illustrated by the use of the data in table 4.3.

Load	Power Rating	Rank	Before Shedding	After Shedding	Required Status
1	6kW	500	on	on	off
2	6kW	900	on	off	off
3	6kW	800	on	off	off
4	14kW	1000→200	off	off	on
5	3kW	100	on	on	on
6	7kW	505	on	on	off
7	15kW	1000	on	off	off

Table 4.3 : Load Ranking Priority

Consider the scenario that an outage has occurred and the maximum supply capacity is reduced to 17kW. Before shedding started, loads 1,2,3,5,6 and 7 were engaged giving a total power consumption of 38kW. The load shedding rules shut down loads in the order of their priority switching out loads 2,3 and 7 thus reducing the demand to 16kW. If some conditions now change which cause the priority of load 4 to change from 1000 to 200, this load now takes priority over loads 1 and 6. These must be shut down reducing demand by a further 13kW to allow load 4, which takes 14kW, be engaged. The total consumption will then be 17kW. Such a situation could occur for a change in occupancy, area function, equipment status or supply capacity. The rule-base deals with this by initially looking for loads which are consuming power and shutting them down until the demand is just less than the supply as described previously. When this has been completed, the *Load_shed* hypothesis object takes on the value false.

Pattern matching is used then to identify all loads that are currently turned off but which require engaging, i.e. the value of their *rank_actual* property is less than 900, or if the BMS is running in economy mode less than 800. A calculation is performed to sum the power requirement to engage these loads.

```
If   SUM[|PLANT|.power_measured] ≥  Electrical.supply_capacity_set
  & Load_shed is False
  & is |PLANT|.control_status = 0,-1
  & is |PLANT|.rank_actual < 900
⇒  Change_ranks is confirmed
  & SUM[|PLANT|.power_predicted]is assigned to Electrical.power_required_actual
  & reset Load_shed
  & do Load_shed
```

```
If   SUM[|PLANT|.power_measured] ≥  Electrical.supply_capacity_set
  & Load_shed is False
  & is Building.mode_status = 'Economy'
  & is |PLANT|.control_status = 0,-1
  & is |PLANT|.rank_actual < 800
⇒  Change_ranks is confirmed
  & SUM [|PLANT|.power_predicted] is assigned to Electrical.power_required_actual
  & reset Load_shed
  & do Load_shed
```

The load shedding rules are then reset and evaluated again which forces the system to shed further loads to make supply capacity available to engage higher the priority loads. This continues until equilibrium is reached for the second time. This rule has a lower inference priority which ensures than it is called after all the load shedding rules are either fired or found to be false.

The engage rules seek to turn loads back on which should be on until the maximum power demand has been reached. For example, if the IBMS is running in economy mode, the following rule re-engages lighting in an incremental way similar to the incremental load shedding.

```
IfChange_ranks is True
  & Load_shed is False
  & Building.mode_status = 'Economy'
  & |PLANT|.rank_actual < 800
  & |PLANT|.class_set = 'Lighting'
  & |PLANT|.control_status < 1
  & SUM[|PLANT|.power_measured] + |PLANT|.power_predicted ×  1.5
      < Electrical.supply_capacity_set
  & MIN[|PLANT|.rank_actual]
  & |PLANT|.light_level_set < Lighting.min_level_set - 20
⇒  Engage_loads is confirmed
  & |PLANT|.light_level_set - 20 is assigned to
      |PLANT|.light_level_set
  & -1 is assigned to |PLANT|.control_status
  & reset Lighting_ranking
  & do Lighting_ranking
  & reset Engage_loads
```

In economy mode all loads with ranking less that 800 are re-engaged. Pattern matching is used to find the sub-set of lighting plant equipment objects which have a rank less than 800 and are currently switched off or partially shed. In each case, a check is made to see if the engaging of this load will exceed the current power supply capacity. The property *power_predicted* stores the average running power consumption the last time the load was engaged. A 50% margin is allowed for error in the normal power demand and for starting surges. This is a figure that may need to be set at different values for different classes of plant. For example, the starting current for induction motors is much higher than their current consumption for normal running speed. Finally, the set of objects is reduced to the single object with the lowest value in its *rank_actual* property. This load is re-engaged.

In this way the higher priority loads are turned on following the increased supply capacity being made available by the load shedding rules. It also ensures maximum use is made of the available power. If one assumes that load shedding has cut some loads to reduce the power demand to 10kW below the supply capacity and that the highest priority load which has last been shed is 12kW. If there exists a 4kW load which has a lower priority and was also shed the engage rules will not re-engage the 12kW load but will go on to engage the 4kW load even though its priority is lower. This occurs because the check for power consumption comes before the minimum ranking order clause in the rule conditions.

The set light level is increased until 80% of its set value is reached. A second rule for lighting re-engagements sets the plant status to 2 when the level -1 control is increased to above 80% of the set level. In the case of heating loads this switches the plant to level 1 control to prevent saturation of control loop when 80% control is exceeded under level -1 control. When the full power supply returns all loads which were shed, or had their set points reduced, are returned to their normal control levels[28]. When there are no longer any loads at status -1, environmental control is returned to the zone controllers. This is achieved by removing the 'BUILDING' objects from the 'out' mode using four rules, one of which is shown below.

```
if    {PLANT}.control_status ≠  -1
      & Building.mode_status = 'Normal'
      & ||PLANT||.rank_actual > 900
      & |||BUILDING|||.location_no_set = ||PLANT||.location_influence_no_set
⇒     Normal_control is confirmed
      & 'out' is assigned to |||BUILDING|||.mode_status
```

This rule uses exhaustive pattern matching to confirm that there is no plant operating under reduced set point closed loop control, a second independent matching finds the sub-set of plant which should be off for normal (non economy mode). This set is cross-referenced with the 'BUILDING' class to find the areas influenced by these loads. The zone controller's state is set to 'out' for these areas. Similarly, a rule exists to switch the controller to the on state and another two to perform the same functions in economy mode. Once returned to either on or off mode the controllers themselves switch to start or stop based on their occupancy predictors. A hardware feature would have to be included on the zone controllers to manually switch them back to zone control in the event of the communication link being lost between

---

[28] Note: The zone's control level need not necessarily be the same as it was before load shedding occurred as time has passed.

the controller and the IBMS and the controller becoming stuck in the 'out' state after power has been restored. This could be achieved by either a time out feature on the 'out' state or a manual reset button on the zone controller itself.

4.3.4 Condition Monitoring and Breakdown Related Control Rules

The following rules provide information to the operator on abnormal environmental conditions and possible faults. In such situations, and where it is possible, the rules take action to compensate for a breakdown by utilising reserve equipment.

4.3.4.1 Condition Monitoring

Rules are included to check for abnormal operation and to signify a fault. Some building management systems already incorporate condition monitoring. For example, many air handling systems incorporate pressure sensing devices to detect fan failures or filter blockages. These signal fault alarms to the IBMS operator. This condition monitoring can be extended by the inclusion of classes of rules for each plant type to check for correct operation.

The condition monitoring is divided into one of two classifications. The first is the reporting that a definite item of plant equipment is thought to be at fault; this is detected by either plant level condition monitoring with sensors or by a failure to respond to control signals[29]. The second classification is when a problem is detected and where it could be due to a any one of a number of causes; this could be the identification of an environmental problem such as an area is too hot, too cold or that the light level is below the desired brightness. The maintenance rules within the control rule-base identify the plant failure or problem and report it to the operator. Further rules, which form part of the facilities management rule-base, interact with the IBMS operator to assist in diagnosing the cause and deciding what action to take.

One of the most annoying features of computerised control systems which was reported to the author by facilities managers at Brunel and Lloyd, is the incorrect, or over enthusiastic, reporting of error messages. If one is continuously bombarded with messages about every minor discrepancy one is liable to miss an important error message. Care must therefore be taken when writing rules to ensure that the hypothesis will only be confirmed if the evidence supporting the problem is sufficiently high. To this end, the condition clauses should test as

---

[29] This does not necessarily mean that the fault is due to the plant as it may be caused by the associated control system.

many properties as can be considered appropriate to increase the confidence in the hypothesis and also to include a margin for sensor error and plant ageing. The problem identification rules are written for each area of the building whilst the condition monitoring is plant specific. A simple example of a problem identification rule is given below.

```
if   |BUILDING|.mode_status is 'start'
     & time_measured > 30 +  |BUILDING|.start_time_predicted
⇒    Condition_monitor_heating_problem  is confirmed
     & report 'not made preheat time in area(s) |BUILDING|.description_set'
```

The rule finds the sub-set of 'building' objects which have been in the pre-heat mode for thirty minutes longer than predicted. It is likely that there is a heating problem in these areas as the set temperature has not been reached. The rule makes use of the *BUILDING.description_set* property to report the problem to the operator. The rule below checks that the correct lighting level is maintained in all occupied areas of the building.

```
if   |BUILDING|.mode_status is 'on', 'start'
     & [|BUILDING|.light_level_actual - |BUILDING|.light_level_set] > 5
⇒    Condition_monitor_lighting_problem is confirmed
     & report "Light level in area(s) |BUILDING|.description_set is incorrect"
```

A five percent margin is included for sensor control loop limitations. The luminaire ageing itself will be compensated at the lighting design stage and during operation by the closed loop light level control.

4.3.4.2 Breakdown Related Control

In the event of a breakdown, the IBMS can seek to compensate for the failure by calling on reserve equipment or modifying the control of related equipment. For example, the failure of a luminaire can be compensated for by increasing the light level of surrounding luminaires by a small fraction until the faulty tube is replaced. In the event of a tube starter problem, or dying tube, this can be shut down to avoid the annoying slow flashing which often occurs. In the case of an air handling unit supply fans the associated extractor fans could have their duty cycle or speed increased to compensate. If a boiler fails to start, reserve boilers could be utilised or the control system could relax the air quality and use more circulated air, thus overriding the air quality control in an attempt to reduce the heat loss. These rules require location specific knowledge which is contained in the plant equipment data-base. For example, in the case of a lighting failure, this will be detected by the local light sensor as described in the previous section. The following rule uses the hypothesis of the condition monitoring rule as its first condition clause and then compensates for the failure by increasing the light level of a related luminaires, or blocks of luminaires.

```
if    Condition_monitor_lighting_problem is True
      & |BUILDING|.mode_status is 'on', 'start'
      & MAX [|BUILDING|.light_level_set - |BUILDING|.light_level_actual]
      & ||PLANT||.class_set = 'Lighting'
      & ||PLANT||.location_influence_no_set = |BUILDING|.no_set
⇒     Breakdown_compenstion_lighting
      & |BUILDING|.light_level_set + 5 is assigned to |BUILDING|.light_level_set
      & reset Condition_monitor_lighting_problem
      & reset Breakdown_compenstion_lighting
```

The rule uses the maximum function to find the single object representing the area at which the light level is the greatest value below the set value. The first clause allows for a five percent error margin. The set of lighting objects from the 'PLANT' class which light the area are also found. The value of the *light_level_set* property is increased for all these objects and both the compensation and condition monitoring rules are reset. This enables the rules to be called again increasing the light level further in this area or another area.

In many cases, due to the design of the building services, no reserve equipment will be available. To make maximum use of these rules a certain amount of redundancy could be built into the services at the design stage where this is cost effective to so do.

4.4 Implementation

A prototype zone controller incorporating an occupancy predictor was built using a stand alone 8086 processor card with digital and analogue input/output to process the data. This was linked to a 486 PC, as shown over-leaf in Fig.4.18, which operated as the central control computer running the global control rule-base. In addition to occupancy prediction, the controller uses an internal mathematical thermal model of the area to predict optimum start and stop times depending on environmental conditions. The hardware design information for the prototype and its communication links with the 486 computer are outlined in Appendix D.

The zone controller could be mass produced cheaply by the use of a general purpose interface and processing chip. Such devices are becoming available at low cost with widespread applications in the field of process control [83]. For integrated systems, where a single building local area network is used such as BatiBus or the German FND, the controller can be linked to the network. Occupancy was counted using simple infra-red break beam type detectors. Depending on the direction of motion the two light beams are cut in a different order and thus the sensor can detect if the occupant is entering or leaving an area. In practice

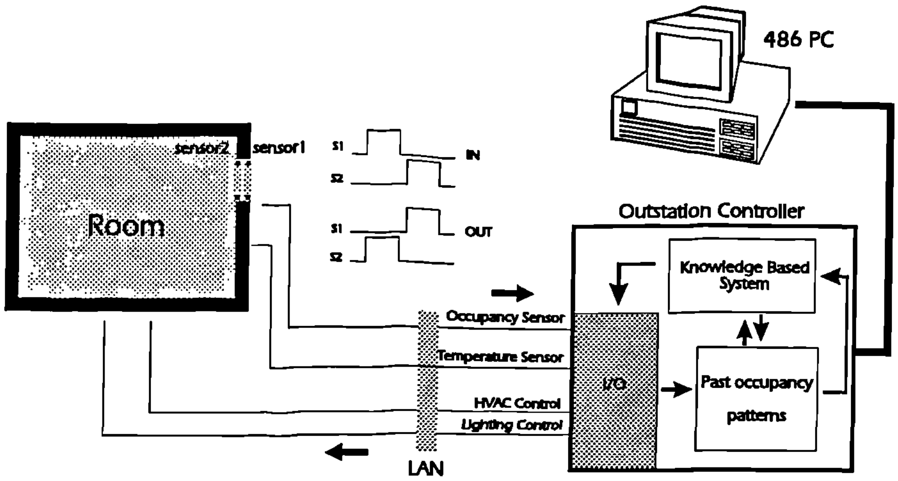


Fig.4.18 Occupancy Sensor

such a sensor would probably have to be used in conjunction with a wide angle motion detector as a safe guard against the counting routine giving a false reading.

In practice, the global control system would communicate with many zone control outstations. As this was not practical, the rules were tested using a software simulation using data which was supplied by the estates department at Lloyd's. The implementation of this simulation is outlined in Appendix D. The rules process data on the Lloyd's heating, electrical and lighting systems. The knowledge-base is used to simulate the control of electrical loads during normal and abnormal supply conditions. This includes setting on and off cycles and load shedding during supply outages. Whilst mainly considering the electrical loads here, the proposed methodology is also appropriate for other systems such as oil fuel boilers or the transportation systems.

4.5 Summary

This chapter has proposed some rule-based control strategies for processing on both the global control system and the outstations. The aim is not to replace or to mimic conventional control techniques with a rule-based system. Instead, the proposed rules are integrated with conventional systems to provide enhanced and more intelligent control features. A unique aspect of the approach is the use of people counting data and activity prediction. This additional information is used by the rule-based system to schedule services to better fit the occupancy in the building and to prioritise services in the event of power shortages.

The use of a simple rule-based system for zone energy control has many advantages. The use of occupancy prediction can provide energy savings by ensuring that the desired environmental conditions are only maintained when the building is in use. This saves the operator's time in re-programming the control schedules for different areas when their function changes. The zone controller 'learns' changes and adapts automatically whilst being intelligent enough to realise when its own prediction may be in error. The built in thermal model allows the controller to predict optimum start and stop times for different environmental conditions. When this prediction is in error, the controller can diagnose the most likely cause and adapts the thermal model accordingly. The controller also includes rules to prevent excessive wear on plant equipment, which overcomes a drawback of many optimum environmental control systems.

The rules written for the central control system include global control strategies and facilities management support. The control rules divide into one of five categories: area security, area control, energy management, supply failure and equipment failure. Sets of rules and rule-based control procedures have been identified for different classes of equipment. These rules override the outstations control rules. The correct environmental conditions are maintained in parts of the building such as wash rooms and corridors that are not continuously occupied by linking their control to related areas of the building. The outstation's occupancy predictor can be overridden by the global control system in the event of a planned event. An economy mode of operation has been defined in which the set points are relaxed in sparsely occupied areas or for out of the normal occupation. The rule-based energy management is combined with traditional load balancing techniques to reduce peak supply consumption. The load control study illustrates how the IBMS structure, with pooled information between services, can be used to reduce overall running cost. In the event of a supply failure, load shedding occurs by prioritising services and making the best use of the available power. Such control features are only possible if all loads within the building are under the direct digital control of the building management system. The main aim of such strategies is to cause minimum inconvenience to occupants.

The ideas have been investigated using a number of case studies. A prototype rule-based outstation controller was built in conjunction with a people counting sensor. This was installed in a large room at Brunel and linked to the existing Trend heating control unit. This enabled the self-tuning aspect to be investigated along with testing the functionality of the outstation rules. The concept of a self-tuning outstation attracted interest from the Brunel estates department.

The next chapter will show how this control system, in conjunction with the object-oriented data-base, can be integrated with the building's facilities management functions.

CHAPTER FIVE

FACILITIES MANAGEMENT APPLICATIONS

5.0 Overview

This chapter explores the role of rule-based systems to provide facilities management functions within an IBMS. Such functions include assisting an IBMS operator in diagnosing operational problems, identifying faults and scheduling maintenance when monitoring the BMS. Existing BMS user interfaces and control room procedures are reviewed and their limitations are discussed. This chapter proposes improvements and defines the template for an improved operator environment. The role for a knowledge-based system to provide operator assistance is outlined in a series of case studies. Faults are diagnosed to a system and then to a component part of that system. A self-learning methodology for fault prediction and diagnostics is proposed using an artificial neural network. Finally, the work is extended to the optimum scheduling of preventative maintenance.

5.1 User Interface for Operator Procedures

5.1.1 Current user interfaces for BMS

There is currently no standard approach to screen layouts for building management systems. A non consistent screen layout can even exists within a particular system. At Lloyd's the maintenance management database system is text based only whilst the HVAC, lighting and lift control systems are each controlled by two screens: one text and one low resolution graphics. The lighting control system is particularly frustrating to use and is much disliked by the engineering staff. To turn on or off a single light requires going through a very long series of hierarchical menus and knowing many access codes. The systems are inflexible and non changeable. In general, the after sales service from the supplies is often poor or very expensive. A supplier may charge an expensive fee to make minor changes. Once supplied, the customers are often forced to accept the problems or change their systems[30].

IBMS manufacturers are now moving very quickly towards fully graphical displays. Two such suppliers are Trend with a Windows based environment and Honeywell with a detailed

---

[30] The manufactures of the maintenance management database system charged Lloyd's around £12,000 just to modify one of their programs to keep track of spare parts. Lloyds are currently planning to change their CALMS (computerised Lloyd's maintenance system) to another preventative maintenance package primarily due to the inflexibility of their supplier.

graphics display. What is required is a consistent approach to graphical user interfaces (GUIs) across the entire system and the ability for the users to make minor changes to the system themselves thus adapting it to the changing requirements. The GUI also needs to adapt to changes in the building and the plant database. The Windows graphical environment seems to be the way all commercial software is moving. An IBMS GUI in a Windows environment allows it to be easily integrated with other windows software such as word processors and spreadsheets. This would be useful for report writing and data analysis. The arrival of Windows NT will soon enable software to be portable across platforms. This would allow a Microsoft Windows system, running on the PC, to run in the X Windows environment for UNIX on a Sun Sparc Station[31].

5.1.2 Proposed GUI Structure for System Operator

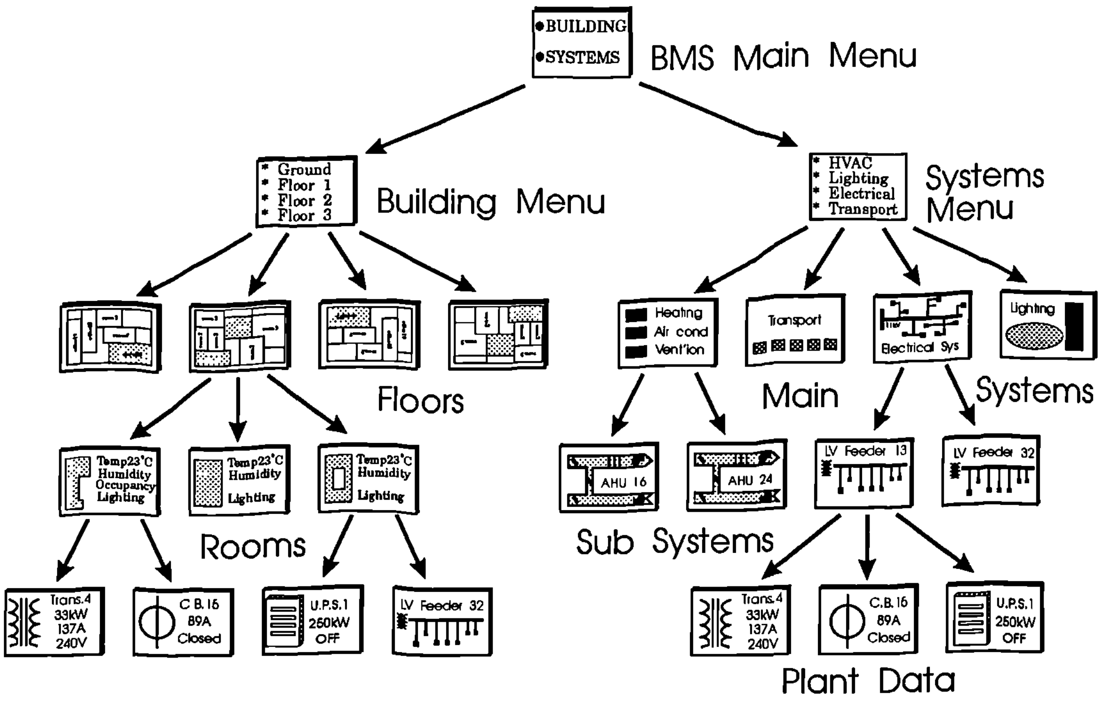


Fig.5.1 Proposed IBMS Operator's Menu Structure

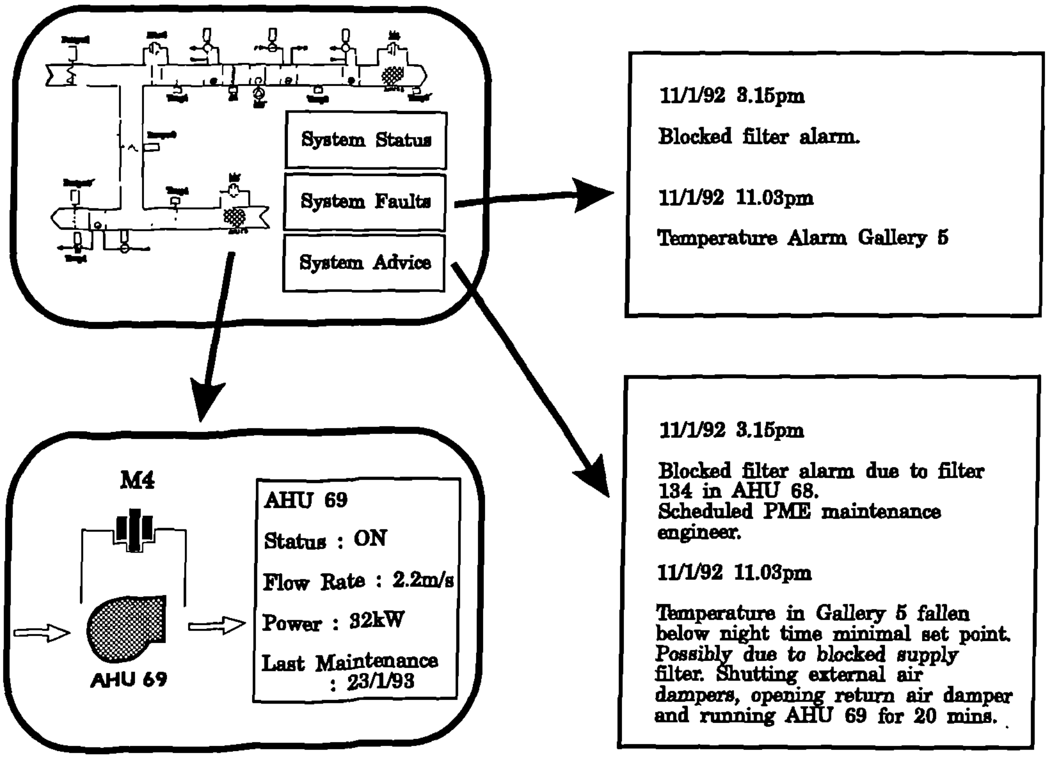
What is required is a GUI which will allow the user to examine the building's systems from an engineering point of view as well as the area they influence. Fig.5.1 illustrates this point by showing a tree structure for the different screen layouts. At the top menu the operator can select the building or the systems menu. If the problem or enquiry is known to relate to a known area of the building, the operator can call up a display for the specific floor and then

---

[31] This feature is claimed by Microsoft for Windows NT however specialised packages already exist to convert Windows 3.1 software for X Windows use.

select from this the specific room screen. If the problem or enquiry relates to an engineering system, one can select the appropriate system from the global menu and call up the system's overall mimic diagram such as the general MV/LV electrical schematic or the elevator car locations plan. From here the operator can further select sub-systems. For the case of HVAC this could be a specific air handling unit.

Such an environment would help to assist the operators in their work. For example, to attempt to restart a chiller one can follow the right hand side of the tree into the HVAC sub-menu and then down to the cooling system sub-menu. From this one can locate the chiller and call its menu options up on the screen along with its control parameters. To change the lighting in an area, one can follow the left hand side of the tree into the floor menu and then select the specific room menu. Here lighting can be selected which switches the menu system to the part of the lighting sub-system associated with this room.



11/1/92 3.15pm

Blocked filter alarm.

11/1/92 11.03pm

Temperature Alarm Gallery 5

11/1/92 3.15pm

Blocked filter alarm due to filter 134 in AHU 68.
Scheduled PME maintenance engineer.

11/1/92 11.03pm

Temperature in Gallery 5 fallen below night time minimal set point. Possibly due to blocked supply filter. Shutting external air dampers, opening return air damper and running AHU 69 for 20 mins.

AHU 69
Status : ON
Flow Rate : 2.2m/s
Power : 32kW
Last Maintenance : 23/1/93

Fig.5.2 BMS Operator's Screen for HVAC System

It is proposed that each engineering screen should contain a diagram along with three text information windows as shown in Fig.5.2. Each item of plant equipment on the sub-system display is represented by a graphics object[32]. By clicking on the plant equipment diagram with the mouse pointer one can then call up relevant engineering data such as the status, power consumption and last maintenance period.

The first text window should display the system status, i.e. whether it is 'on' or 'off', the power level, the error from set point, etc. The second gives a list of faults and problems along with the time that they occurred. This uses the information generated by the condition monitoring rules described in chapter four. This window would be a scroll bar type[33] which would allow the user to look back over past faults. The final window displays an explanation for the current system status and operator advice in the event of a fault. The text in this window is generated by rules which process both current and historical data. This chapter outlines a methodology and procedure for providing knowledge-based advice for the operator in this third window.

The HVAC systems were chosen for a case study and three areas were investigated in the field of knowledge-based operator procedures for building management systems. HVAC was chosen as data was readily available from the Lloyd's CALMS and Johnson's systems. The first area considered is in assisting the operator in handling a problem reported by the systems themselves, i.e. an alarm situation from the HVAC system stating that a boiler will not start. By asking the operator a series of questions based on symptoms of past problems the fault is diagnosed to a particular item of plant equipment. The emphasis is on the optimum inspection sequence rather than diagnosing the fault instantly. The IBMS also gives the operator an approximate repair time and suggests the class of maintenance engineer most suitable to handle the problem. The system will attempt to give reasons for the breakdown and identify trends and related problems.

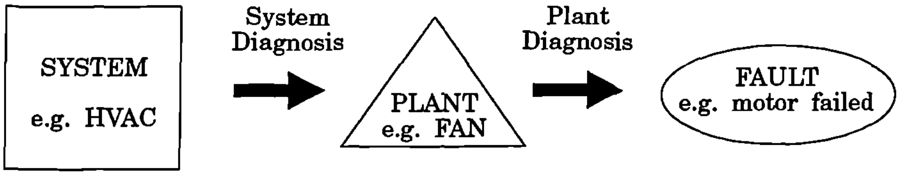


Fig.5.3 Overall System/Plant Diagnosis Methodology

The second rule-base handles the situation when a problem is detected by someone in the building. For example, an occupant may be in an area with a HVAC fault which causes it to be too warm or too humid. They telephone the control room or maintenance department and

---

[32] In terms of computer graphics an object is an individual graphic element which has properties and a defined behaviour. e.g. a graphic button is a object with properties size, colour, location and up or down.

[33] A scroll bar window is a text windows with an effective size larger than its actual size on the screen. The user can move backwards to see text which has scrolled off past the top of the window.

speak to the BMS operator. The knowledge-based system invites the operator to answer a series of questions and diagnoses the problem firstly to a sub-system and then to a specific item of plant equipment. This can be considered as system level diagnosis whilst the first rule-base operates at plant level diagnosis [Fig.5.3].

The final section of this chapter proposes a self-learning methodology to predict fault probabilities, maintenance repair classes and times. This prediction is based on historical breakdown data and uses an artificial neural network. In the case of certain components the probability of failure is linked to many variables. For example, running time, outside temperature, time of the year, use of associated equipment, age, breakdown history, etc. It is proposed that an artificial neural net is trained with all these variables which can be obtained from the breakdown history database and used to predict failure probability. In the case of HVAC there are several hundred breakdown history cases which can be used for net training. The work extends to investigate the possibility of self-learning fault prediction methodology for system wide diagnosis. This leads on to the optimisation of preventative maintenance. The aim of a preventative maintenance program is to aid in minimising the unanticipated failure of equipment in order to avoid the resulting down time and subsequent loss of operation. This also minimises the cost of emergency repair and possible hazards to personnel. If carried out correctly it can maximise the efficiency of the equipment and prolong its useful life.

Current maintenance management systems are very much an isolated entity and do not form part of the BMS. There is no ability for direct connection to any of the building's control systems for automatic data capture and update. The final section in this chapter proposes that the maintenance system should be packaged as part of the overall IBMS such that the scheduling of maintenance can take place based on analysis of historical breakdown and costing data. This could be used to optimise the scheduling of preventative maintenance and thus reduce the likelihood of breakdowns.

5.2 Fault Diagnosis

The fault diagnosis system helps the operator to diagnose a problem based on alarms and fault codes reported by either the control rules or information from the service engineers. This could be accessed by the BMS operator in radio contact with an engineer. The maintenance engineers could even be issued with notebook computers and connect to the building's main LAN from the source of the problem. This would provide them with access to all the maintenance and diagnostic data from the central IBMS computer to assist them in their work.

Fig.5.4 shows a diagrammatic overview of the inference process. The fault is diagnosed by evaluating the status of symptoms and alarms whilst establishing that other related faults are false. Forward chaining then occurs to conduct pattern matching on related past breakdowns. The results of this search are then used by the trend analysis rules to give further advice to the operator. The action could be to remedy a fault, to find out more information about a fault or to carry out some modification of adjustment.

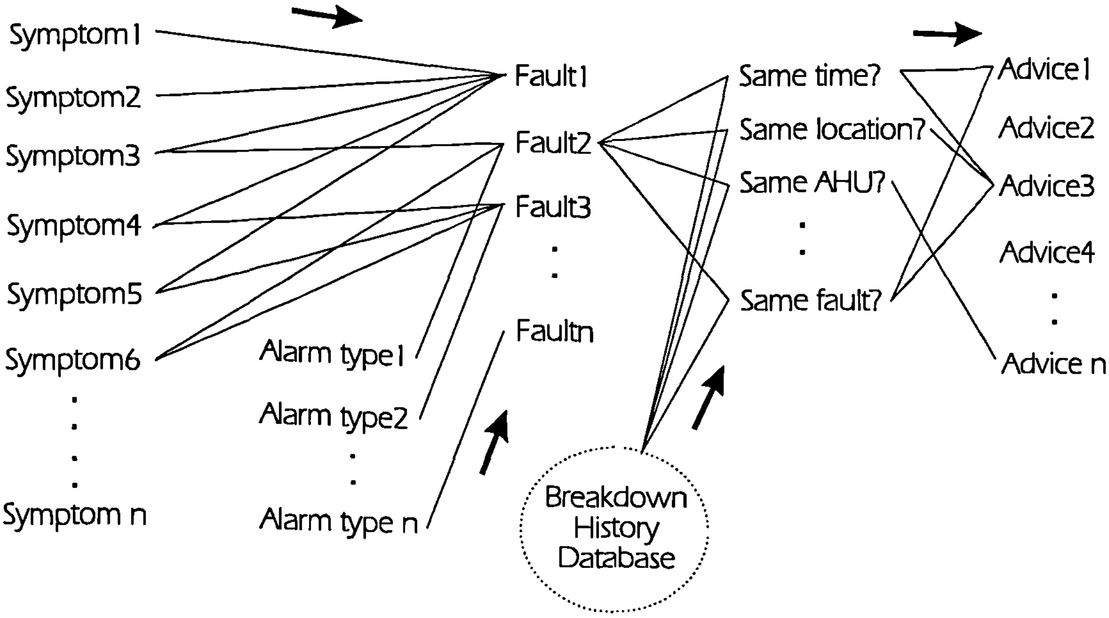


Fig.5.4 Overview of Rules for Fault Diagnosis

The control system rules described in chapter four inform the IBMS operator if equipment is not operating as expected. For example, the boiler lockout rule will become true after the boiler has tried to start a few times but failed. Further rules then diagnose that this could be caused by a lack of ignition, gas or a water level gauges reading too low. In the case of no ignition this would be sensed by a thermocouple heat sensor and the gas inlet would be turned off automatically. In the event of an excess temperature alarm the cause could be a stuck gas valve, low water or low water pressure. The symptoms are stored as objects, some of which are listed in Table 5.1. Each symptom has a single property which can have the values 'True', 'False' or 'Unknown'. A set of symptoms are written for each fault which take the form of questions for the operator to answer. A symptom can be related to more than one sub-class of plant equipment. For example, an excess temperature alarm may occur for a chiller or a boiler.

Symptom no	Description
8	"Has the boiler been re-adjusted for clean combustion?"
8a	"Is the smoke from the boiler too dark?"
8b	"Is the problem due to the Johnson's controls?"
9	"Is the pressure incorrect?"
10	"Has the lockout relay failed?"
10a	"Has a fault occurred?"
11	"Is there a problem with the controls?"
12	"Does the flow valve need adjusting?"
13	"Is there a problem with the shut down point?"
13a	"Is the boiler running ?"
14	"Does the boiler need to be put into 'Summer mode operation'"
14a	"Is a seasonal adjustment required?"
15	"Do boilers need to be put back on gas?"
16	"Are the boilers running in auto?"
16a	"Are the boilers running in manual?"
17	"Has an alarm occurred?"
18	"Has the smoke density meter been cleaned recently?"
19	"Do the boiler rod tubes need to be removed for inspection?"
20	"Do the boilers need to be changed from gas to oil?"
21	"Does the gas booster need resetting?"

Table 5.1 : Symptoms for Boiler Fault Diagnosis

Class	Fault	Description	Trade	Likelihood	Hours
boiler	1	adjust mixture to give clean combustion	pme	1.54	4.00
boiler	2	boiler alarm	pme	15.38	4.24
boiler	3	boiler controls	pme	4.62	10.50
boiler	4	boiler lockout alarm	pme	36.92	0.96
boiler	5	boilers not running in auto	pme	6.15	1.50
boiler	6	check operation	pme	1.54	0.58
boiler	7	clean boiler smoke density meter	pme	1.54	2.00
boiler	8	control problem	jcs	1.54	2.00
boiler	9	excess temperature alarm	pme	3.08	0.88
boiler	10	incorrect pressure	pme	1.54	1.00
boiler	11	lock out relay failure	jcs	1.54	3.00
boiler	12	oil select status point alarm	pme	1.54	2.50
boiler	13	open\close flow valve	pme	1.54	1.00
boiler	14	Investigate (reset) the problem with shut down point	pme	1.54	4.00
boiler	15	put boiler into service in summer mode	pme	1.54	4.50
boiler	16	put boilers back on gas	pme	1.54	2.00
boiler	17	remove boiler rod tubes for inspection	pme	1.54	17.50
boiler	18	request to change boilers from gas to oil	pme	3.08	4.50
boiler	19	reset gas booster	pme	1.54	0.41
boiler	20	shut down boilers for other maintenance	pme	3.08	0.79
boiler	21	smoke density alarm	pme	6.15	0.63
boiler	22	warm through ready to put into winter mode	pme	1.54	2.75

Table 5.2 : Fault Objects for Boilers

The system first asks the user what class of equipment the fault has occurred on. The user is invited to choose an option from a menu which includes air handling units, boilers, chillers, coolers, fans, heat exchanges, lighting, pressure units and pumps. The operator is then asked a series of questions regarding the symptoms of the fault with answers 'yes', 'no' or 'do not know'. A powerful feature of the rule-base is the ability to reason with unknown data as well as definite facts. The system then selects the most likely fault description and what action to take. It also recommends which class of maintenance engineer to send (i.e. the building's own staff or contract maintenance) and gives an indication of the likely repair time. These last two entries are based on historical breakdown data. The system tells the user where the plant equipment is located in the building and, if applicable, whether it forms a part of another piece of equipment such as an air handling unit. This enables the operator to inform the maintenance engineer exactly where to find the equipment. Rules are then fired to cross reference the breakdown database and give further information to the operator. These rules identify possible trends and give advice to the operator such as "this is the 3rd time a fan fuse has blown on AHU 23 in two weeks, check the motor and bearings". In a busy control room with many maintenance engineers, and control room operators working shifts, such trends could easily be overlooked.

5.2.1 Diagnosis Rules

Table 5.2 shows a list of twenty-two fault objects from the fault database for the boilers. Although called faults, the work may be anything which requires the maintenance's staff time. This does not necessarily mean correcting a fault as preventative maintenance work falls into this heading. For example, to shut the boiler down for other maintenance (fault code 20) means the boiler must be temporarily shut down to maintain some other part in the heating system. A problem may be due to the controls and not the boiler itself. Such a problem is dealt with by the control engineering contractors and time should not be wasted sending a boiler engineer to service this problem. Some fault codes represent actions to be taken following a fault or checks to be made to prevent an incorrect diagnosis.

One of the first condition clauses for most of the rules is to test symptom10a which ascertains if the required work is to remedy a breakdown or if it is for some other reason. If it is not 'False', i.e. 'True' or 'Not Known', the fault diagnosis hypotheses are placed on the agenda. Further tests are then conducted to see of the plant is running and if an alarm has occurred.

```
If   Input.class_set is "boiler"
     & Symptom10a is not False
     & Symptom17 is True
     & Boiler_alarm is "excess temperature alarm?"
     & Fault10 is False
     & Fault13 is False
⇒    Fault9 is confirmed
     & Fault9.fault is assigned to Input.fault
```

The rule shown above diagnoses a temperature problem with the boiler. The 'Symptom10a is not False' clause is used to include the possibility 'Notknown' as the value of Symptom10a. There are many symptoms that may not be deemed to be due to a fault. For example, in this case the boiler will probably be running so an inexperienced operator may answer 'no' to the question 'is there a fault?'. The rule checks the status of hypothesises *Faults10* and *Fault13*. This causes backward chaining to these rules to ensure their hypotheses are false. If the excess temperature alarm has gone off and the fault is not due to excess pressure or the flow valve does not need adjusting, the problem could be due to an excess temperature alarm fault. This may mean a failure of the thermostat or sensing circuitry. For *Fault9* to be confirmed the problem must be due to the circuitry associated with the boiler and thus is a job for the boiler engineers. The full set of rules for boiler problem diagnosis, which form part of the diagnosis rule-base, are annotated in Appendix F. Similar sets of diagnosis rules exist for all main classes of plant equipment. The rule-base's primary function is to decide which class of engineer to send and to prioritise the urgency of the problem rather than diagnose the fault in detail. The urgency factor is used later in the optimised maintenance management rule-base.

Each diagnosis rule is written for a different class of equipment. The first left hand side condition is usually a test on the *input.class_set* property. This value is entered by the IBMS operator. This enables the rule-base to be self contained for a particular class. New classes of plant can be added and rules can be changed within a class without effecting the operation of the system. This leads to a modular approach to the rule-base construction. However, the situation could occur that the operator may think that a fault is due to a particular class of equipment but in fact it is caused by another. This then causes the incorrect set of diagnosis rules to fire. Rules have therefore been written to enable the diagnosis procedure to change between rule classes.

```
If   Input.class_set is "Air handling Unit"
     & Symptom39a is not False
     & Symptom39b is not True
     & symptom17 is True
     & AH_alarm is "fan tripped alarm?"
⇒    Fault45 is confirmed
     & report "This problem is therefore due to the fan and not the AHU itself."
     & 'fan' is assigned to Input.class_set
     & Fault55 is True
```

The rule shown above diagnoses a problem for the class air handling units. If the temperature in the area is incorrect and the 'tripped fan' alarm has gone off the sub-goal *Fault45* becomes true. The right hand side action writes a text message on the screen and the *input.class_set* property is changed to 'fan' which causes backward chaining to the fan diagnosis rules. Similar rules are written for other plant to switch inference to sub-parts or related plant.

The order in which the rules are called is based on their inference priority. For each of the diagnosis rules the inference priority is set to the fault's likelihood property. This means that when backward or forward chaining is not prevalent, the fault diagnosis inference occurs in the order the faults are most likely to occur. Similar sets of rules have been written for the other classes of plant equipment. As the reply options to the symptom questions include 'Not Known' the situation could occur that the questions are answered so vaguely that no definite diagnosis could be reached. A rule is included to take account of this. In the event of no other diagnosis being reached with some of the symptoms having the 'Not Known' value in their status property, a text string is displayed telling the operator to find out more information on those specific symptoms and to try again. One further rule is included to take account of a fault or diagnosis not included in the rule-base. If all answers to symptoms are either true or false and no fault is diagnosed the diagnosis is recorded and this is added to the rule-base.

In general, the approach to writing a new rule can be summarised by the flow diagram shown in Fig.5.5. An action can either be in response to a fault or not. If it is a fault it may be triggered by an alarm. If this is so, conditions are added to test the symptoms associated with the fault. A distinction is made for faults which always cause an alarm and those which may cause an alarm. For example, the flow valve may need adjusting which may cause the excess temperature alarm to occur. This symptom could be triggered by an excess pressure or temperature detected in another part of the heating system and suggested as a boiler fault by a change of class rule. The faults where an alarm may occur are written in such way that they force backward chaining into the faults where the alarm must occur by checking that the fault objects for related faults are false. Care must be taken not to form a recursive loop when writing these rules where fault x may only true if fault y is false and vice versa. If an alarm must occur for a fault to be true the rules are written such that the symptoms for all faults where this alarm may occur are tested thus placing these related fault rules on the agenda. This has the effect that if an alarm occurs, rules for faults associated with this alarm are fired first even though a test for the alarm itself does not exist in these rules. This therefore allows for the possibility of second order effects of a fault causing the alarm and thus prevents an erroneous diagnosis.

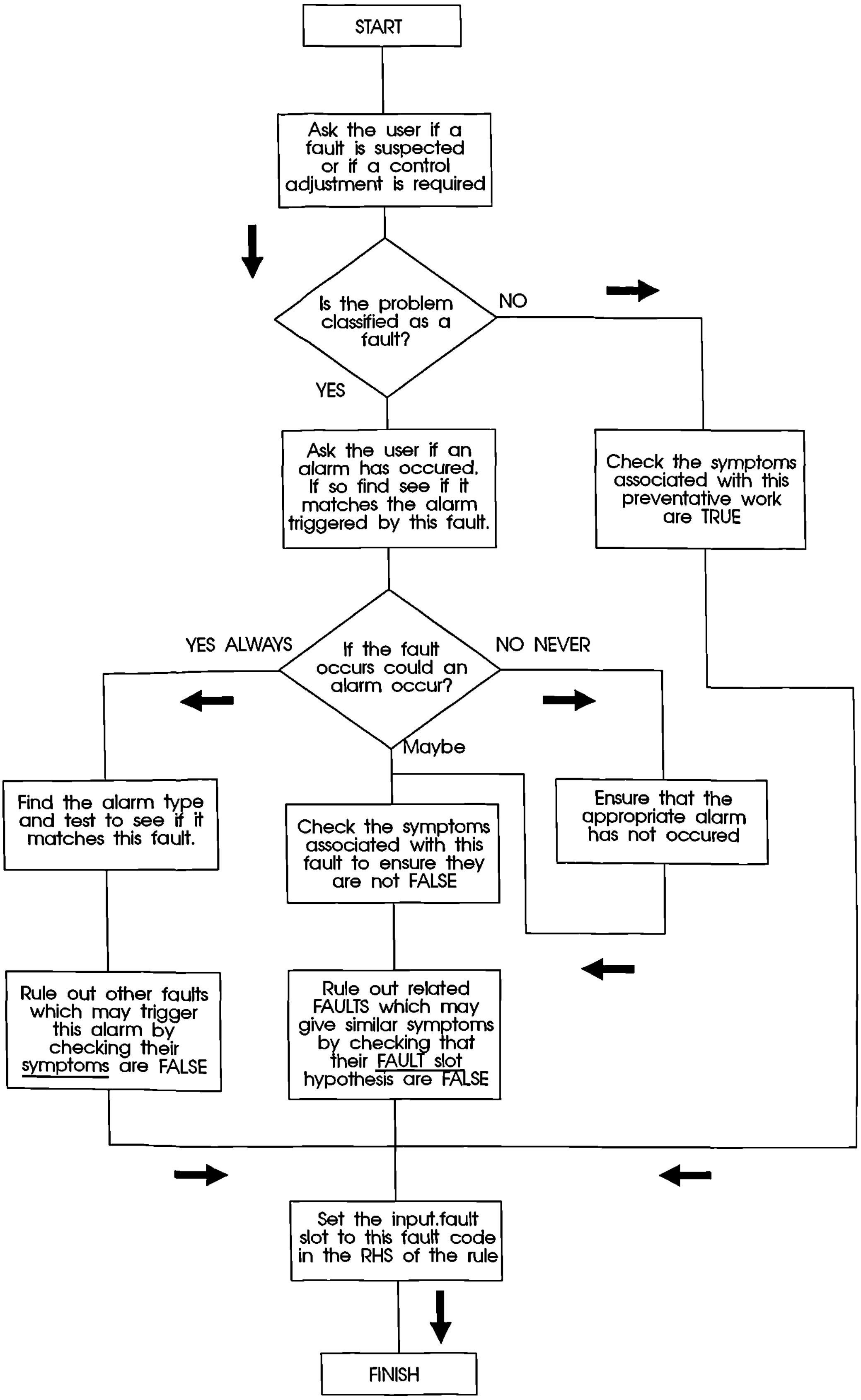


Fig.5.5 Flowchart for Diagnosis Rule Construction

5.2.2 Advice and Trend Analysis Rules

When the diagnosis is reached there is a check to confirm that the fault code has been evaluated. The system then asks the user for the specific identification number for the faulty equipment. The hypothesis *Check_input2* backward chains to this rule whilst checking that the plant number exists for that item of equipment thus validating the input. The location and functional description of the equipment are retrieved from the HVAC data-base and displayed to the operator in a message window. The operator thus knows where to send the engineer to investigate the problem.

```
If   Input.fault_set is greater than 0
     & |||Hvac|||.class_set is equal to Input.class_set
     & |||Hvac|||.no_set is equal to Input.no_set
⇒    Find_location is confirmed
     & |||Hvac|||.location_no_set is assigned to Input.location_no_set
     & |||Hvac|||.ahu_set is assigned to Input.ahu_set
     & report  Input.class_set " number" Input.n_set "is located in" Input.description_set
          "and forms part of" |||Hvac|||.function_set
```

Once the fault or action has been established, and advice on repair time and class of engineering staff has been given, the rule-base system then proceeds to give further advice based on previous breakdown history and trend analysis. Rules are fired which cross reference the breakdown history database and gather information. The results of this search are then used by a further set of advice rules.

Rule Hypotheses	Search Criterion	Reason
Trend_Sp	same plant	to get data on specific plant reliability
Trend_Sc	same class	to get data on general plant reliability
Trend_Spsf	same plant same fault	to see if this equipment is especially prone to this fault
Trend_Slst	same location same time	to diagnose location related faults
Trend_Sfsc	same fault same class	to diagnose a recurring fault for a specific plant type
Trend_Sfst	same fault same time	to diagnose a time related problem for PM
Trend_Spsfst	same place same fault same time	to diagnose a specific plant problem

Table 5.3 : Summary of Past Breakdown Search Criteria

Table 5.3 summarises some of the search rules. The advice rule-base contains a class of rules whose hypothesis objects start with 'Trend_S' which stands for trend search. The rule with hypothesis *Trend_Spsf* which stands for 'search plant same fault' is given on the next page.

```
If  ||BREAKDOWN||.class_set is equal to Input.class_set
    & ||BREAKDOWN||.no_set is equal to Input.no_set
    & ||BREAKDOWN||.fault_set is equal to Input.fault_set
⇒   Trend_Spsf is confirmed
    & ||BREAKDOWN||.period_set is assigned to Trend_Spsf.period_set
    & ||BREAKDOWN||.fault_set is assigned to Trend_Spsf.fault_set
    & ||BREAKDOWN||.hours_set is assigned to Trend_Spsf.hours_set
    & SUM(||BREAKDOWN||.hours_set) is assigned to Trend_Spsf.thours_set
    & SUM(||BREAKDOWN||.n_set) is assigned to Trend_Spsf.n_set
```

The rule conditions conduct pattern matching on objects belonging to the 'Breakdown' class. If the class and number properties match the item of plant, the sub-set of objects is reduced to faults which have occurred on the same plant. This set is further reduced by pattern matching on the breakdown fault properties to give just a list of the same faults for this specific class. If some objects exist in this sub-set, the hypothesis is proved true and the rule actions assign specific data from the search into properties belonging to the hypothesis object. In this case, for the last occurrence of this fault on this plant, the maintenance period and repair time are all recorded. The total time taken up repairing this fault and the total number of breakdowns for this class of equipment are calculated using the 'SUM' function. This data is used for later analysis.

Some faults cross reference more than one data class. For example, the rule with hypothesis object *Trend_Sfsl* is,

```
If  ||BREAKDOWN||.fault_set is equal to Input.fault_set
    & |||Hvac|||.location_set is equal to Input.location_set
    & |||Hvac|||.class_set is equal to ||BREAKDOWN||.class_set
    & |||Hvac|||.no_set is equal to ||BREAKDOWN||.no_set
⇒   Trend_Sfsl is confirmed
    & ||BREAKDOWN||.Period_set is assigned to Trend_Sfsl.period_set
    & ||BREAKDOWN||.fault_set is assigned to Trend_Sfsl.fault_set
    & SUM(||BREAKDOWN||.hours_set) is assigned to Trend_Sfsl.hours_set
    & SUM(||BREAKDOWN||.n_set) is assigned to Trend_Sfsl.n_set
    & SUM(||BREAKDOWN||.hours_set) is assigned to Trend_Sfsl.thours_set
```

This search looks for the number of times the same fault is occurring at the same location. Pattern matching is conducted on the breakdown objects to find the sub-set of objects which match the fault code. A separate pattern matching is conducted on the HVAC database to find the set of plant in the same location. Both sub-sets are then matched together to find the set with the same class and plant number. Fig.5.6 outlines this search using a Venn diagram.

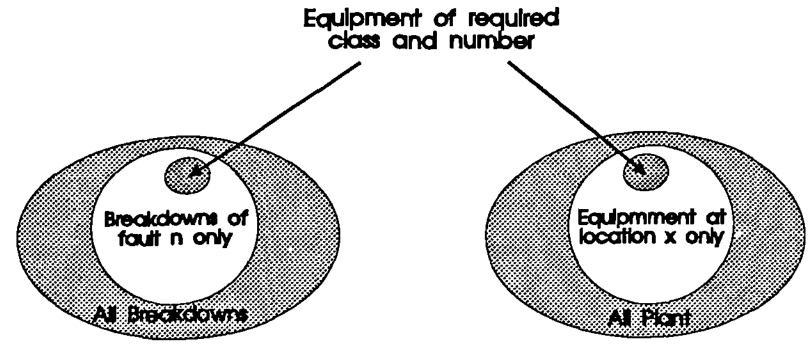


Fig.5.6 Venn diagram showing pattern matching for the *Trend_Sfsl* Rule

The search rules report to the operator any pertinent information resulting from the search. This is intended as background information only. For example, in the case of routine adjustments such as a change over from gas to oil, the system will tell the user when the boiler was last put onto oil fuel. In the case of a boiler rods inspection, the system will give the date of the last inspection. Other rules will tell the operator how many times this fault has occurred on this particular plant and as a percentage of the total number of breakdowns, e.g. "Blocked filter problems account for 70% of all faults on this AHU". Rules also determine if a fault has never occurred on a particular plant before.

The next set of rules look for trends in the breakdown data. An example of this is to inform the operator if a particular fault is occurring more often that it should on a particular plant. This decision is made by the following inference,

If *Trend_Spsf.n_set + Trend_Sp.n_set - 0.1 > Trend_Scsf.n_set + Trend_Sc.n_set*
⇒ *Too_frequent is confirmed*
& report "This fault is occurring too often for this plant"

The total number of breakdowns caused by this fault on this class of plant is stored in the property *Trend_Spsf.n_set* and the *Trend_Sp.n_set* property stores the sum of all breakdowns irrespective of fault code for this plant. *Trend_Scsf.n_set* and *Trend_Sc.n_set* store similar results but for the whole class generally. If the percentage occurrence of this fault is greater than 10% of its occurrence for the class in general, the fault is defined as occurring too often. The inference makes the assumption that the probability of one fault occurring more than any other should be the same for all plant within the same class. This does not necessarily mean that the probability of breakdown for a particular item of plant is the same for all plant of the same class. For example, some plant may get more use than others[34]. Further rules are written of a similar nature and are summarised in table 5.4.

---

[34] This could also mean that one fault is more likely than another if the useage patterns are different. This this fact is not taken into account but could be included.

Rule Hypotheses	Uses Searches	Reason/Advice
Trend_B2	Trend_A2,Spsf	To tell the user how recently the same fault occurred
Trend_B3	Trend_A3, Spsf, Sp	The fault is occurring more than any other fault on this plant
Trend_B4	Trend_A4, Sfsc, Sc	The fault is occurring more than any other fault on this class
Trend_B5	Trend_Spsfst	The fault is occurring too often on this plant
Trend_B6	Trend_A6, Sfst	The same fault has occurred too often this period

Table 5.4 : Summary of Search Inference Rules

For each of these rules, once the hypothesis is confirmed a text string is constructed which is written to the message window on the screen. In the case of the rule below, interpretation[35] is used to obtain the fault description from the fault object as well as the plant item's class. These rule force backward chaining to the search class of rules.

```
If    Trend_A4 is True
      & Trend_Sfsc.n_set ×  2 - Trend_Sc.n_set is greater than 0
⇒     Trend_B4 is confirmed
      & report fault\Input.fault_set\.description_set "is occurring
         more that any other fault on the" Input.class_set
```

The hypothesis is proven by the condition *Trend_Sfsc.n_set* × *2 - Trend_Sc.n_set > 0*. This is tested to see if the occurrence of this specific fault on this class is occurring more than any other fault in this class.

The final sets of rules in this rule-base draw on the conclusions of the search and inference rules to advise the operator on fault trends. The hypothesis object name for each of the rules begins with "*Trend_C*" and the number which follows corresponds to the number of the related fault object. Table 5.5 summarises some of these rules and they are discussed in more detail below.

Rule Hypotheses	Uses Searches	Reason/Advice
Trend_C4	Trend_B5	Give reasons why the lockout alarm keeps occurring
Trend_C10	Trend_B5	Give reasons why the temperature alarm keeps occurring
Trend_C21	Trend_B5	Suggest cleaning of the smoke density alarm sensor
Trend_C38	Trend_B6	Suggest PM on filters on this air handling unit
Trend_C39	Trend_B5	Possible electrical fault as numerous fuses keep blowing
Trend_C64	Trend_B2	Diagnose possible nitrogen leak

Table 5.5 : Fault Specific Trend Analysis Rules

---

[35] Interpretation is a means of accessing a dynamic objects name by constructing it from the values of other properties.

For example, the rule with the hypothesis *Trend_C4* is proved true if the fault is signalled by a boiler lockout alarm concurrent with the problem occurring too often for this particular plant as defined by hypothesis *Trend_B5*. If this is true, it could be due to a problem with the ignition circuit, the gas valve, a low water level or a fuel problem. Similarly, if rule hypotheses *Trend_C10* and *Trend_C39* become true, the right hand side gives specific advice for the particular reoccurring fault such as the fuses blowing.

Some rules use the *Trend_B6* hypothesis with a specific fault code. For example, in the case of hypothesis *Trend_C64*, if the nitrogen bottles need changing and the were only changes 3 weeks ago this could suggest a possible leak as they normally last much longer. Hypothesis *Trend_C38* becomes true if the same fault is occurring too often this period in a specific class of plant as diagnosed by the rule with hypothesis *Trend_B6*. If the filter is blocked this could suggest that it could be a good time to conduct preventative maintenance and change other filters in this air handling unit.

This set of rules could be extended by utilising pattern matching by physical location of equipment. For example, a fault on an item of equipment at the same time as a false fire alarm reading can be used to diagnose the cause of the miss alarm as the equipment overheating.

5.3 System diagnosis

The second rule-base deals with a problem reported to the BMS control room by one of the occupants. A diagnosis is made to a system or sub-system within the building. The rules first locate the problem to a specific air handling unit finding the location where the HVAC conditions are reported to be incorrect. The system displays a list of floors within the building and the user selects the appropriate one followed by the area where the HVAC conditions are reported to be incorrect. The system cross references the location with the HVAC data-base and informs the user the number of the air handling unit associated with this area for both the air supply and extraction. The user then answers questions on symptoms appropriate to this area and air handling unit. The problem is then diagnosed to a class of plant equipment. The rules cross reference the symptoms and class of equipment with the HVAC fault record database. A list of likely items of plant equipment are given with the plant location and reasons for why they have been chosen. This allows the control room operator to contact the maintenance engineers, tell them what equipment to look at, what the problem might be and exactly where to go. After gathering more data from the plant equipment, the first rule-base is used to diagnose the actual fault on the equipment.

5.3.1 System Diagnosis Rules

The inference tree structure for the problem diagnosis is shown in Fig.5.7. The inference process is initiated by placing the hypothesis *Problem* onto the agenda. This causes backward chaining to the *Find_ahu* rules which read in the location of the reported problem. Forward chaining then occurs to decide which class of equipment could be the cause based on the symptoms. Further rules are then used to suggest the plant items to be checked.

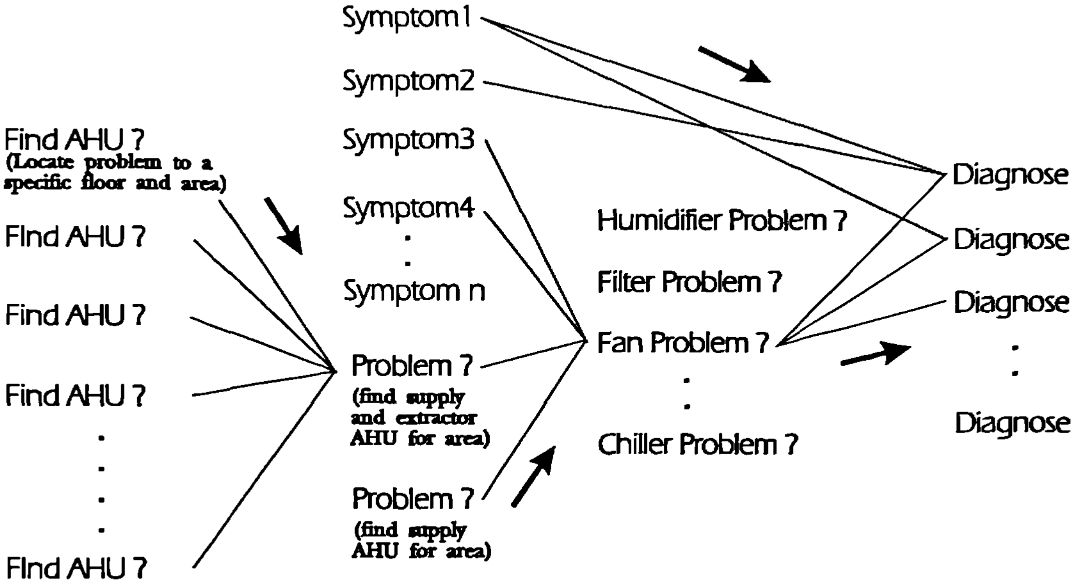


Fig.5.7 Inference Structure for System Diagnosis

Once the air handling unit has been identified, the system forward chains to rules which diagnose the class of plant equipment which could be the possible cause of the problem. The rules attempt to identify the class of plant most likely to be causing the problem and then give a list of plant equipment numbers.

If Problem is True
& Symptom201 is True
⇒ Problem_filter is confirmed
& report "The problem may be caused by a blocked filter in this AHU."

For example, the rule shown above uses *Symptom201* to diagnose a possible blocked filter problem. The symptom for this is the area being stuffy and lacking air movement. Similarly in the rule shown over-leaf, *Symptom 202* is used to diagnose a humidifier fault.

```
If   problem is True
     & Symptom202 is True
⇒    Problem_humidifier is confirmed
     & report "The problem may be due to a humidifier in this AHU."
```

5.3.2 Plant Class Diagnosis Rules

If the problem is diagnosed to a class of equipment there may only be one of these associated with the particular air handling unit. Most units only have one humidifier and main supply fans. The plant number suggestion is then very trivial, simple including pattern matching on the 'HVAC' class to the specific unit. However, in the case of filters, there may be quite a few in any one air handling unit. In this case, use is made of the maintenance data to make an informed 'guess' at which filters are most likely to have become blocked. This includes some knowledge of the operation of the plant, its function within the system and in the case of filters the time since last maintenance was last conducted.

```
If   Problem_filter is True
     & ||Hvac||.class_set is "filter"
     & ||Hvac||.ahu_set-Input.ahuS_set is precisely equal to 0
⇒    D1 is confirmed.
     & 1-MIN(  Hvac |.last_maintenance_actual) is assigned to output.weeks
     &   Hvac||.no_set is assigned to output.no_act
     & ||Hvac||.location_set is assigned to output.location_set
     & report "Check filter number " output.no_act " on AHU " Input.ahuS_set ,"
        which is located at " output.location ". The maintenance records indicate
         that this filter has  not been changed for " output.weeks " weeks."
```

In the case of filters, the rule looks for those for which maintenance has not been conducted for the longest time. This uses pattern matching on the maintenance database. A range of values are suggested and in each case the location is given for the engineer to service the equipment along with what symptoms to look for to confirm the fault.

This rule-base thus provides the maintenance engineer with a list of equipment to check and what symptoms to check for. The equipment's location is given along with any special maintenance information such as the requirement of special tools or parts. The knowledge-base design is modular in construction and class oriented. Classes of diagnostic information can be added or modified without effecting the operation of the inference diagnosis. This allows for easy maintenance and expansion.

5.4 Fault Prediction

For the final part of this work an investigation was made into combining fault prediction with the fault diagnosis. This would give a diagnostic system which could diagnose faults based on symptoms as well as past break down history. If trained regularly, the system would become self-tuning in the same way as the human operator learns on the job. This is an advantage over the KBS approach as, providing a set of symptoms and faults have been defined, modification would require no rule programming.

When gathering the breakdown data from Lloyd's it was noted that problems on equipment often occur in patterns such as a number of blocked filter alarms occurring on the same air handling unit during the same maintenance period and then none for several months. Another observation was that a series of alarms of one type would occur, the equipment would be inspected and no real fault found. After a series of such events, the equipment would breakdown. In complex systems a fault in one area can place an extra strain on components in another area and cause further problems. An investigation was carried out to see if it is feasible to train an artificial neural network to predict the probability of breakdown based of the recent fault history and other factors. The investigation used fault data from the Lloyd's air handling units but any other system such as the hot water system or transportation could have been used.

The work assumes that different air handling units may be susceptible to different fault probabilities as they can be subject to different usage patterns and may be constructed differently. This could mean that the fault probability and repair time may both vary for different air handling unit systems at different times of the year. Fig.5.8 shows graphically the breakdown statistics for various air handling unit fault codes and the average repair time for the faults is shown in Fig.5.9. Both graphs represent twelve months of cumulative breakdowns for the year ending spring 1991 The repair time is an important factor in determining if it is worth while conducting preventative maintenance. As can be seen in the figure, one occurrence of fault code 47 is as significant in labour cost terms as many code 44 faults. Appendix F gives a full list of defined fault code and symptoms numbers.

Working on the hypothesis that faults can be related, and past events may influence the fault probability for specific plant, an artificial neural network predictor was investigated. A neural network application package called NeuralDesk was used to investigate this theory. NeuralDesk runs on a PC under Windows 3.1 and integrates with the Excel spreadsheet database. The common standardisation of the Windows 3 environment makes it very easy to integrate these with the Nexpert Object KBS shell and Excel data-base. Appendix G gives an

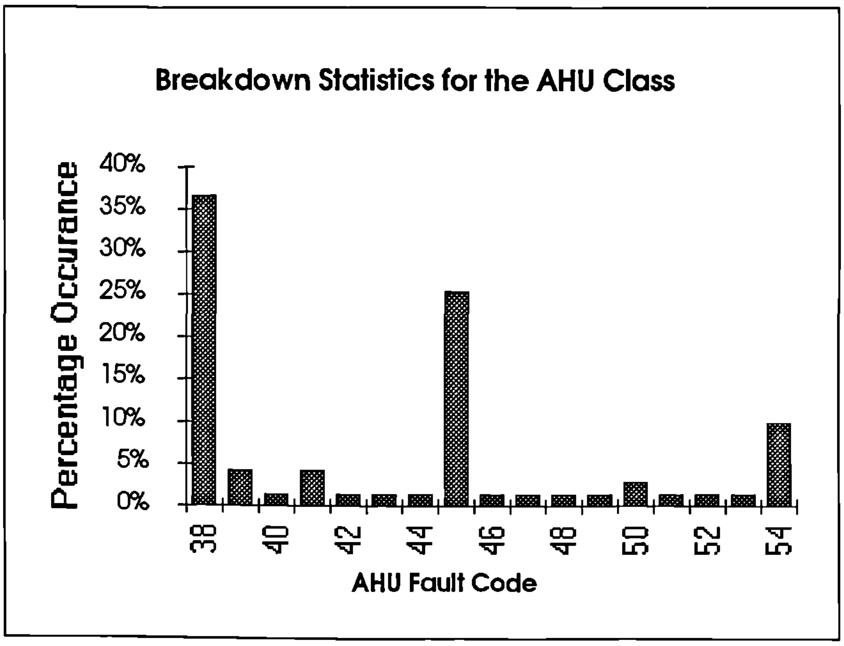


Fig.5.8 Fault Breakdown Statistics
(Reproduced by kind permission of Lloyd's of London)

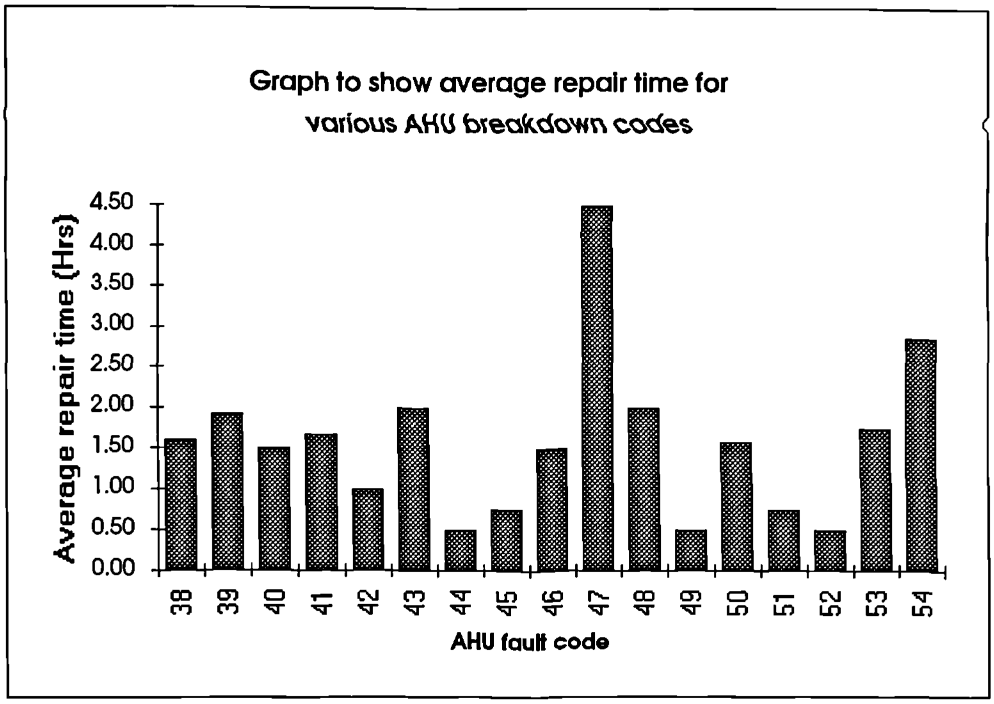


Fig.5.9 Fault Maintenance Time Statistics
(Reproduced by kind permission of Lloyd's of London)

overview of neural networks and outlines the implementation details of NeuralDesk. The proposed structure is shown in Fig.5.10

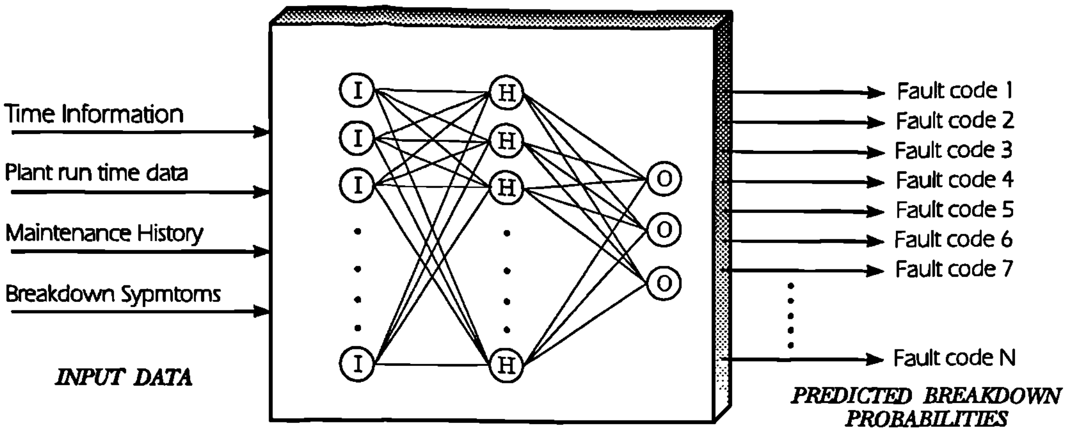


Fig.5.10 General Maintenance Data Breakdown Probability Predictor

Different approaches were tried to establish the if the network could find sufficient correlation in the breakdown data to make fault prediction feasible. The network configurations and training results are given in Appendix G. The final approach adopted is illustrated in Fig.5.11. The network fault inputs take account how recently the faults occur. If a breakdown occurred three weeks ago, the fault input stimulus is set to the reciprocal of the whole number of weeks. The significance of the fault input thus decreases with time. If the same fault occurs again in the same period this input is set to one again. The networks were trained using a year's breakdown data from the Lloyd's CALMS system. Most of the data was used for training but some was reserved for interrogating the network.

The symptoms are entered on a scale or symptom certainty from 0 to 1 as outlined below.

$$o_i = \begin{cases} 1 & \text{If the symptom is present} \\ 0.75 & \text{If the symptom is probably present} \\ 0.5 & \text{If it is not known if the symptom is present} \\ 0.25 & \text{If the symptom is probably not present} \\ 0 & \text{If the symptom is not present} \end{cases}$$

If no knowledge is known about any symptom the symptom training input is set to 0.5. The above definition thus allows a wider range of uncertainty over the symptoms and therefore the scope for a more accurate diagnosis. When a fault is correctly diagnosed its symptoms are recorded and this data is used as training vectors for the network. This is a much simpler procedure than writing rules. The class of air handling units was again chosen for investigation and the diagnosis rules for fault objects 39 to 54 were re-coded in the above form. The air handling unit diagnosis rules use 18 symptoms and 4 alarm signals.

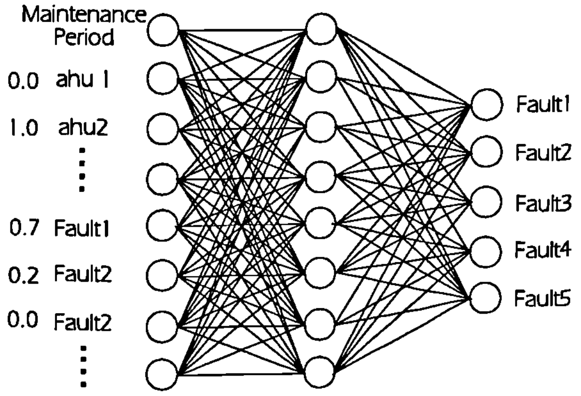


Fig.5.11 Fault Diagnosis Network

The back propagation algorithm was chosen as this gave the lowest prediction errors. The network was trained to predict the fault code, the repair time (as a fraction of 3 hours) and the class of maintenance engineer (1 = Contract maintenance, 0 = Lloyd's staff). The results are shown in table 5.6.

Fault 5 (300 epochs) BP (0.9,0.1)	**Mean error for fault data**	
	Training	**Prediction**
single layer 6 neurons	0.077104	0.068320
2 layers 6 neurons	0.232818	0.110624
1 layer more neurons	0.072781	0.048941
1 layer 6 neurons feed forward	0.064374	0.046313
1 layer more neurons feed forward	0.066167	0.047235
2 layers 6 neurons feed forward	0.064247	0.047466

Table 5.6 : Fault Prediction Based on past Breakdowns and Symptom Knowledge

The full training and interrogation results are given in Appendix G. As can be seen in table 5.6, the single layer network with feed forward gives the lowest overall training and prediction errors. For this network, the average fault prediction error was 2.9%, for maintenance classification it was 7.7% and the average error for repair time prediction was 8.1%. In the case of the maintenance trade, the value was rounded to zero or one before the error value was calculated.

A very low training error was not expected as there are inconsistencies in the data. As can be seen in the tables in Appendix G, in all cases that the back propagation algorithm produced the lowest training results. The general conclusion is that the lowest training error is obtained by keeping the network simple. Little if any improvement is gained by a large number of hidden neurons or more than one hidden layer. The more layers of hidden neurons the network has increases the training time. It can be seen that extending the network to two hidden layers significantly increases the training error. However, the feeding forward of data from the input to the output layer reduced the training error in each case. The lowest training error was obtained with a single layer network with minimal neurons and feed forward links. About one fifth of the available fault data was kept for interrogating the network to test its prediction ability using this network. In each case the error turned out to be slightly below the training error.

This work was conducted as a feasibility study and requires further work using a greater range of test data than was available at the time. A further step would be to integrate breakdown data from different classes of equipment across different systems. The network could establish links between problems within the building which span different systems. Such possible links include the loss of electrical power causing alarms, a boiler failure causing loss of heat elsewhere or a fan problem causing a temperature level alarm in another area of the building. This could be further enhanced by adding a wider range of input factors such as the outside temperature, weather and equipment running-time information.

If the diagnosis system were to be implemented using a neural network it would have the advantage of becoming self-tuning as new training data becomes available. This is unlike the rule based approach where once they are written the rules cannot easily adapt by changing their inferences. The neural network does have the disadvantage that adding new faults or symptoms requires the re-training of the entire network. Therefore, a compromise solution has been presented which uses the best features of each whilst avoiding some of their respective problems.

5.5 Optimised Maintenance Management

The previous sections considered rules for diagnosing faults and problems. This section explores the role of a rule-based system in scheduling maintenance resources and in particular, optimised preventative maintenance.

Many buildings already have a maintenance scheduling program for breakdown and preventative maintenance work. As outlined in chapter two, there is a requirement to 'tune' this scheduling to best fit the requirements of the building. As these requirements change with time there is a need for a continually adaptive system especially in the field of preventative maintenance. The aim is to maximise equipment reliability and minimise maintenance, repair and replacement costs by the optimum use of man power and resources. A rule-based approach is proposed and the concept is illustrated by the use of a case study for the lighting systems using data from the Lloyd's 1986 building.

5.5.1 Preventative Maintenance

In most planned maintenance systems the maintenance interval for equipment is fixed and can only be changed by the user. Ideally the planned maintenance should be set to best match the requirements of the plant throughout its life cycle. The building's use and requirements will change with time. Hence the maintenance work and the priorities allocated to such work should also change. By using a rule-based approach to this 'intelligent tuning', the system can also explain why maintenance is scheduled at a particular time by the use of engineering rules, actual breakdown data and statistical analysis. This would be of particular use to the maintenance and engineering staff within the building who may not have faith in 'computer judgements' without knowing the facts behind the decision making.

For finite maintenance resources the rules can optimise maintenance scheduling on the basis of priorities. If two separate pieces of maintenance work should be undertaken during the same time period and there is insufficient resources to undertake both of them the KBS can re-schedule one of them based on economic or other factors and then explain its reasoning. The KBS can also be used for predicting breakdowns, maintenance cost and spare part requirements. This is particularly useful for planning a budget request and ordering materials. The system can advise on re-order times and stock of spares based on predicted requirements. It can also advise on specific maintenance requirements following a breakdown, highlight common or particularly costly faults and provide decision support for repair or replacement. A KBS can be used for asking 'what if questions' such as setting constraints and conditions or to find the predicted effect of changing equipment operation. One can use the IBMS as a decision support tool to decide if it would be more economic to reduce or increase maintenance staffing levels, re-distribute staffing levels between different disciplines or to reduce or increase planned preventative maintenance. Rules are written to schedule maintenance based on the following factors:

(a) The probability of a breakdown for a particular piece of plant equipment. This can be predicted by statistical analysis of historical breakdown data from the same class of equipment [84,85]. This can also be linked to the use of the equipment within the building.

(b) Conditioning monitoring using sensors at the control level to monitor the system and signal to the management system when a problem or fault condition is detected. For example, pressure sensors on either side of an air filter within a ventilation system can be used to detect the time to change an air filter. This information is obtained from the global condition monitoring rules given in chapter four.

(d) The cumulative cost of a reduction in performance can be considered. In the case of lighting, an example of this might be the cost of wasted electricity not being converted into light due to dirt build up on the light fittings. This information can be found using a mathematical model for each class of equipment.

(e) The cost of the labour for specific trades can be used to decide if there is an advantage in using different maintenance staff to perform planned maintenance. This might favour investing more money into preventative maintenance or conversely to reduce the overall maintenance cost if one allows for the cost of servicing breakdowns.

(f) The usage prediction for the equipment up to the time of planned preventative maintenance which requires prediction based on the usage data stored within the IBMS control database.

(g) The scheduling of other equipment within a particular class. For example, if a particular piece of equipment is to be serviced at a given time then it would probably be most cost effective to also service other equipment of the same type at the same time.

(h) Scheduling of different classes of equipment with finite resources can take into account the number of maintenance staff, their working hours, the level of spares in stock and the constraints on taking the equipment out of use for maintenance.

(i) Consideration of the equipment's location and site specific factors will allow the access time to the plant equipment to be predicted. Location knowledge also allows the system to schedule maintenance on plant in the same proximity.

(j) Maintenance should be scheduled at an economic 'down times' for a parent class. For example, preventative maintenance of equipment's sub-components can be conducted at the same time. This may avoid having to take it out of service again.

The specific classes of data required for the rule-base have been outlined in chapter three. There exists a plant equipment database which has been divided into six sub-classes namely security, transport, HVAC, electrical supply, lighting and fire prevention. Each of these sub-classes inherits eight maintenance properties.

- A description of the use of the equipment
- A failure probability prediction
- The date of last maintenance (planned or breakdown)
- The cost of planned maintenance
- The planned maintenance period
- The predicted use at the time of maintenance
- An indication of monthly running cost
- The location within the building

Individual sub-classes may have specific properties. For example, florescent lighting has the additional properties of average tube life, the labour rate for a bulk replacement, average light level, minimum light level and luminaire light depreciation constants.

5.5.2 Proposed Knowledge-based Approach

Three different criteria have been identified for writing rules for planned maintenance. The first is the purely economic criteria which schedules planned maintenance based on equating the cost of planned maintenance against the predicted incurred cost of not conducting planned maintenance. The second criteria is concerned with customer satisfaction. This necessitates the introduction of imaginary costs or limits to allow for possible discomfort for the occupants of the building. Thirdly there is are criteria based on finite maintenance resources available such as a fixed budget, staffing levels or spare parts. Each of these areas have been considered.

The structure of the rule-base is illustrates in Fig.5.12. Maintenance is scheduled on the basis of three sets of rules which represent human considerations, maintenance resource constraints and economic considerations. The highest priority rules are the human considerations. These set limits on the number of breakdowns allowed and safety considerations that may effect the occupants of the building. Maintenance resources then take second priority as work cannot take place at a certain time if the staff and materials are unavailable. Finally, the maintenance is scheduled at the most economic time providing these two former criteria are met. This time is calculated by a cost function which uses historical maintenance data to predict failure probability and maintenance costs. Plant usage and wear are predicted using run time data from the plant equipment data-base. This is then used in conjunction with condition monitoring data to predict breakdown probability. The economic decision rules calculate the cost of breakdown maintenance using this information along with the extra cost of the reduction in the operating performance if such a breakdown occurs.

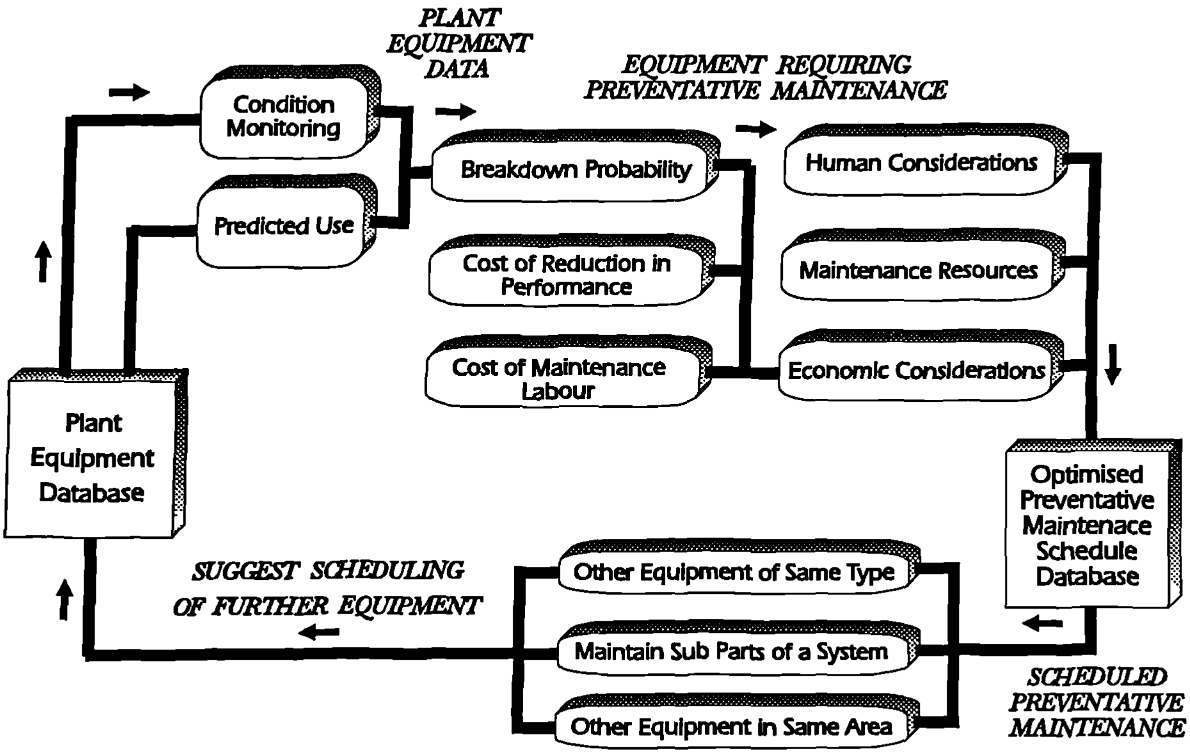


Fig.5.12 Knowledge-based structure for optimised maintenance management

5.5.3 Economic Considerations Rules

There are four economic criteria for scheduling optimised maintenance:

- Preventative breakdown at any cost.
- Preventative breakdown at a given cost.
- Preventative breakdown at minimal cost.
- A combination of the above.

Preventative maintenance at any cost might be employed in situations where the consequences of a component failure are so great that any such failure must be avoided at any cost. An example of this situation might be found in a Nuclear power plant where a failure in the plant equipment could be very dangerous. There are also cases where a failure of one component might cause excessive stress and hence failure of another. This second component might be of far more importance than the first. Therefore, such a component might have an 'effective' cost of failure far higher than its actual cost.

Preventative maintenance at a fixed cost is more common. There is always preventative maintenance work that could be done but which is not possible within the context of a fixed annual budget. Preventative maintenance at a minimal cost is the usual aim. Preventative maintenance is undertaken in the expectation that the cost of such work will be recovered as savings in reduced breakdowns. This therefore forms an equation of maintenance cost against breakdown cost which must be solved to maintain the building at minimum cost. In practice, the maintenance of the building is conducted as a mixture of all three. In most situations there will always be a fixed budget within which as much preventative maintenance work should be undertaken to ensure that the most return is made as savings in reduced breakdown costs. A fixed budget will place restrictions on the resources available to the maintenance department in terms of materials and labour which will restrict the amount of preventative maintenance possible at a particular time. Allowance must be made for breakdown and irregular maintenance work.

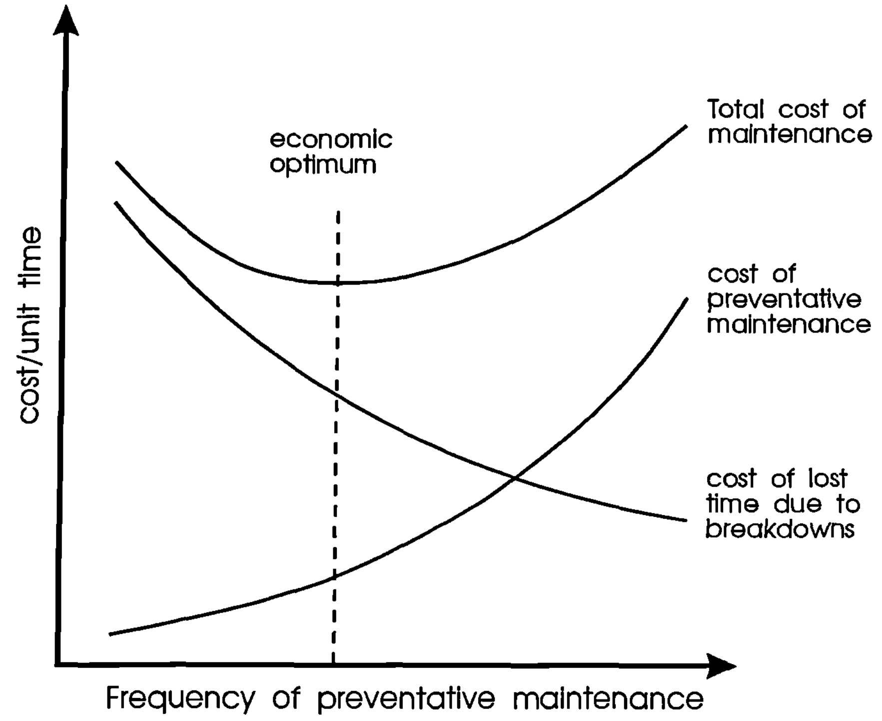


Fig.5.13 Optimum Economic Preventative Maintenance Frequency

The most economic time to conduct preventative maintenance is obtained by means of a mathematical function. It is assumed that the breakdown probability increases with time and that conducting preventative maintenance reduces the probability of a breakdown. If this is so, the more frequently preventative maintenance is carried out the more expensive it is per maintenance interval. If maintenance is conducted less frequently this increases the probability of failure which increases the failure breakdown cost. Assuming that the breakdown cost is greater than the cost of conducting preventative maintenance, there should be an optimum frequency for conducting preventative maintenance. This concept is illustrated in Fig.5.13.

Mathematical models for preventative maintenance have been proposed for may years [86,87] but these do not seem to be used in computer maintenance management systems. Current systems are often just an asset register data-base with scheduling software. The mathematical model requires knowledge of the breakdown probability. Within an IBMS this can be found from actual breakdown data and fitted to known breakdown patterns. This allows the breakdown probability to be calculated outside the known data range.

Many failure-causing mechanisms give rise to measured distributions of times-to-failure which approximate quite closely to probability density distributions of definite mathematical form. These are known in statistics theory as p.d.f.s. Such functions can be used to provide mathematical models of failure patterns, which can then be used in performance forecasting calculations. Some p.d.f.s. relating to maintenance studies include the Negative exponential, Hyper exponential, Normal (or Gaussian) distribution, the Weibull, gamma, Erlang and Lognormal distributions. The four most common p.d.f.s used in failure prediction are summarised below [88].

(a) The negative exponential distribution is one which arises in practice where failure of equipment can be caused by the failure of any one of a number of components of which the equipment is comprised. Also it is a characteristic of equipment subject to failure due to random causes. This distribution is found to be typical for many pieces of industrial plant equipment. The p.d.f. is,

$$f(t) = \lambda \exp [- \lambda t] \quad ..... \quad (13)$$

where λ is the mean arrival rate of breakdowns and $1/\lambda$ is then the mean of the distribution.

(b) The hyper exponential can be used to model failure times which may be very short or very long. Failures at both ends of the time scale are thus greater for the hyper exponential p.d.f. than for the negative exponential. Some electronic equipment such as computers have been found to fail according to this distribution.

$$f(t) = 2k^2\lambda \exp[-2k\lambda t] + 2\lambda(1 - k)^2 \exp[-2(1 - k)\lambda t] \quad ..... \quad (14)$$

where λ is the mean arrival rate of breakdowns and k is a parameter of the distribution.

(c) The Normal distribution applies when the time to failure is the consequence of a large number of small and independent random variations. When this is so the failure distribution is 'bell-shaped'.

$$f(t) = \exp\left[\frac{1}{\sigma\sqrt{2\pi}}\left(\frac{-(1-\mu)^2}{2\sigma^2}\right)\right] \quad ..... \quad (15)$$

where μ is the mean of the distribution and σ the standard deviation.

(d) The Weibull distribution is an empirical distribution which appears to fit a large number of failure characteristics of equipment like electronic components and metals such as steel. The density function of the Weibull distribution is,

$$f(t) = \frac{\beta}{\eta} \times \left(\frac{t^{\beta-1}}{\eta^{\beta-1}}\right) \exp\left(-\frac{t^{\beta}}{\eta^{\beta}}\right) \quad .... \quad (16)$$

where η is a scale parameter (or characteristic life) and β is a shape parameter. When $\beta=1$ the Weibull is equivalent to the negative exponential. If β takes a value greater than about 4 it begins to approximate to the normal distribution.

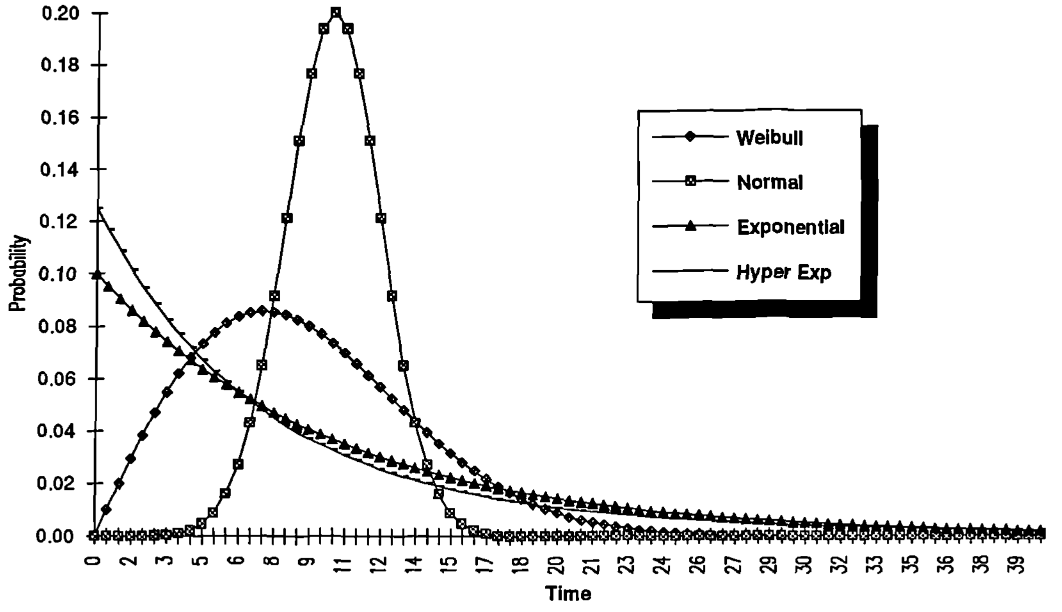


Fig. 5.14 P.d.f.s. Used to Model Failure Characteristics

(from "Models of Preventative Maintenance" by Gertsbakh, ref.88)

The characteristics of these four p.d.f.s are shown together for typical parameter values in Fig.5.14. Combinations of these can be used to construct more complex failure rate models such as the 'bath tub curve' where failures are high during a 'running in' period, then after time they adopt a random pattern and finally near the end of the equipment life they begin to increase due to components wearing out. A p.d.f., or combination of p.d.f.s, is stored as a method to calculate the failure probability for each class of equipment.

5.5.4 Human Considerations Rules

The human consideration rules impose set limits on the preventative maintenance based on conditions and requirements set by the user. They fall into one of three categories:

- Safety factors
- Inconvenience of breakdowns to occupants
- Inconvenience of maintenance to occupants.

The safety factors include preventative maintenance to ensure safe plant operation. For example, the boiler rods must be periodically inspected for signs cracks. A maximum time must be imposed between inspections. The same is true for fire prevention equipment where statuary requirements exist. In some cases a limit may need to be imposed on the allowable breakdown probability to avoid inconvenience for the occupants. A breakdown in the lift system at Lloyd's causes great inconvenience to the underwriters, either by increased waiting times, forcing them to move to another lift tower or even use the stairs. A very low limit for the breakdown probability may wish to be imposed on the Lloyd's lifts thus requiring preventative maintenance to be conducted regularly. This category can also include excessive wear of an item instead of a full breakdown. For example, in time filters become blocked or light output falls. In each case energy is being wasted by the system comprising of worn or non maintained sub-parts. This reduction in performance could be detrimental to the occupants. For lighting the illumination level drops and for blocked filters the air may become stuffy.

The last category of rules ensures that the plant is maintained at a time of least inconvenience to the occupants of the building. For example, a major overhaul of the heating system in the middle of the winter would be unwise. This category also included rules to force plant to be scheduled at a time when a parent item of plant is also undergoing maintenance; i.e. if the boiler is to be shut down and cleaned in a certain maintenance period, the pumps, and associated valves should be checked at the same time.

5.5.5 Maintenance Resources Rules

The available maintenance resources provide yet another constraint which can override the economic considerations. The building must have finite maintenance resources such as a financial budget, a set number of maintenance staff and a sufficient store of spare parts. These factors introduce constraints either on the maintenance period or the maximum number of breakdowns which can be tolerated. When preventative maintenance is considered for a period the maintenance resource rules check to see if sufficient resources are available. The diary data-base is accessed to find all other scheduled maintenance and resources in terms of maintenance time allocated to this period. This is subtracted from the available time calculated from data in the resources data-base to check that this work can be undertaken. The spare part requirements are calculated and checked against the stock level stored in the resources data-base; if these are insufficient the operator is advised via the message log. This enables parts to be bought in time.

5.5.6 Maintenance Rules

Appendix H outlines a case study into the use of a knowledge-based maintenance management within the IBMS for lighting. Lighting is one of the main systems within an IBMS. In a presentation given by Chris Engert [89], the Estates Manager of the Lloyd's 1986 building, the maintenance of the building's lighting system was highlighted as being a major problem therefore lighting was chosen as the initial area to investigate. Also, the Lloyd's 1986 building already has an advanced lighting control system and all the data required to investigate integrating the control system with maintenance is readily available.

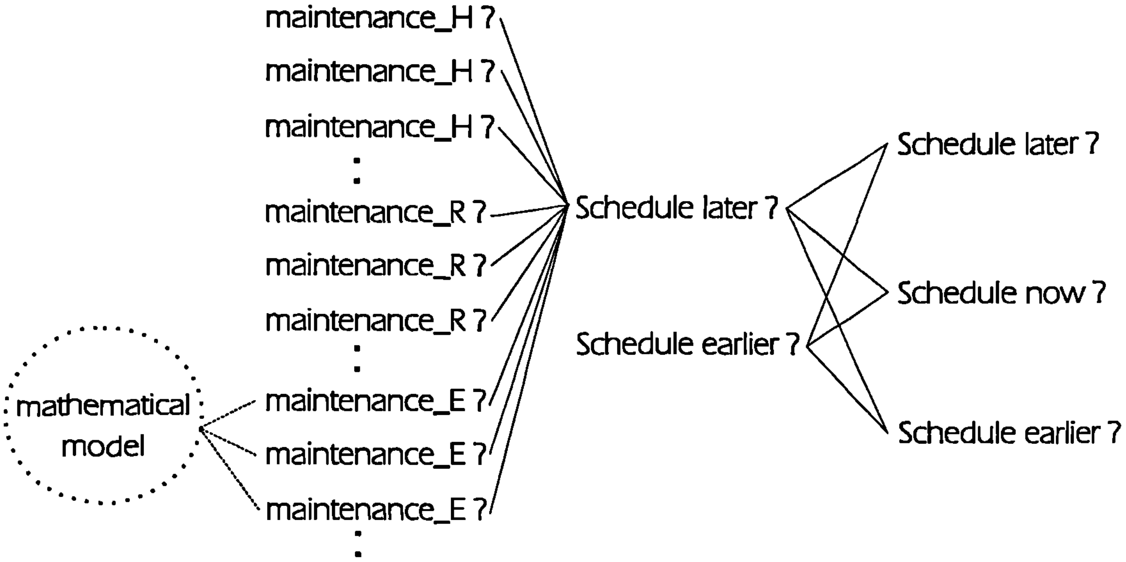


Fig.5.15 Rule Structure for Maintenance Scheduling

The rule inference structure is illustrated on the pervious page in Fig.5.15. Once the plant sub-class and equipment number have been entered the first rule to fire decides if the maintenance time is optimum. This backward chains to two rules with hypotheses *Schedule_later* and *Schedule_earlier.* These rules take the current scheduled period and find if improvements can be made by moving it forward or backwards. If the hypotheses of both these rules are not true, it means that the optimum time has been found. The maintenance period is then updated in the database with the current value of the *|PLANT|.current_schedule_period_actual* property The rule is shown below.

```
if   schedule_later is not True
     & Schedule_earlier is not True
     & |PLANT|.class_set = input.class_set
     & |PLANT|.no_set = input.no_set
⇒    Schedule now is confirmed
     & Create_Object |DIARY|
```

The above rule backwards chains to two rules, one of which is shown below. A dummy value is placed into the property *|PLANT|.new_schedule_period_actual* and the rule forces further backward chaining to rules which check the human, resources and economic factors. A set of these rules is written for each class of plant equipment. A later value is used for the maintenance period in the schedule earlier rules and vice versa. Many rules may have the hypotheses *Maintenance_H* and *Maintenance_R.* Thus any hypothesis rule can cause the left hand conditions to fail. According to rule's inference strategy the economic consideration rules are only processed after all the *Maintenance_H* and *Maintenance_R* hypotheses are known. This saves processing time in unnecessarily evaluating mathematical formulas. This last condition requires evaluation of the *new_maintenance_costs* property which is evaluated by the equation 22 (this is derived later) and is written as a lighting class method. In each case the rule's hypothesis becomes true if there is an economic improvement providing the conditions set by the *Maintenance_H* and *Maintenance_R* rules are met.

```
if   |PLANT|.class_set = Input.class_set
     |PLANT|.no_set = Input.no_set
     |PLANT|.schedule_period_act + 1  is assigned to |PLANT|.new_schedule_period_act
     Maintenance_H is not False
     & Maintenance_R is not False
     & Maintenance_E is not False
     & |PLANT|.new_maintenance_costs_pred < |PLANT|.maintenance_costs_pred
⇒    Schedule later is confirmed
     |PLANT|.new_schedule_period_set is assigned to |PLANT|.schedule_period_act
     |PLANT|.new_maintenance_cost_pred is assigned to |PLANT|.maintenance_cost_pred
     & reset Schedule_later, Maintenance_H, Maintenance_R
     do Scheduled_optimum
```

By definition, only one of the two scheduling rules will be true at the same time as for one of them the economic test condition will fail. In each case they change the value of the *maintenance_costs* property to the value of the *new_maintenance_costs* and reset the effected rule's hypothesis. This forces inference of the maintenance scheduling criteria rule again. The inference continues until both *Schedule_later* and *Schedule_earlier* are false.

For the lighting example, the minimum value to which the light level should fall is usually calculated by the lighting engineer at the design stage. If the cost to conduct a spot replacement is only slightly greater than the individual cost for a bulk replacement, the light level may be very low at the economic maintenance period (<60%). In this case, leaving maintenance until the light output is very low may be unacceptable. The solution is to set a minimum light level beyond which maintenance must occur whether it is the economic time or not. This level may of course vary from area to area. This is an example of a human factors rule.

```
if    lighting_n.light_level_measured ≥  input.min_light_level
      Maintenance_H is confirmed
```

For the Lloyd's data it was found that for a floor of lighting with an average use of less than about 79 hours per week, the light level fell below a perceived minimum level of 60% before the economic maintenance period. For a floor with greater than 80 hours per week use the economic maintenance period was reached first. This proved true using the values of average tube life and maintenance cost supplied by Lloyd's. This data is given in Appendix H.

These rules conduct some simple tests to see if maintenance is possible with the available resources at the proposed time. The first rule to be called which is shown below, checks that there are the required number of maintenance staff hours available for the work.

```
if    |Diary|.period_actual = new_maintenance_period
      |Diary|.reason_set = 'maintenance'
      ||||Lighting||||.location_set = input.location
      |||Fault|||.no = 97
      & SUM[||staff||.time_worked_set] - SUM[|Diary|.hours_set] >
         |||Fault|||.time ×  ||||Lighting||||.n_set
⇒    Maintenance_R is confirmed
```

Pattern matching finds the sub-set of all scheduled maintenance jobs for the period in question. The sub-set of all lighting objects for the area in question, and the fault object for bulk tube replacement, are also found. The rule's condition clauses test that the sum of all available man hours for that period is greater or equal to the predicted time to bulk replace all the lights for this location. Decisions on how the work is scheduled within the maintenance period is left to the IBMS operators or the maintenance foreman.

```
if   |Diary|.period_set = input.period_set
     & ||Lighting||.location = input.location_set
     & |Lighting_spares|.no_tubes_actual < SUM[||lighting||.n_set]
⇒    Maintnance_R is confirmed
     & report "warning, require more spare tubes for this bulk maintenance"

if   |Diary|.period_set = input.period_set
     & ||Lighting||.location_no_set = input.location_no_set
     & |Lighting_spares|.no_tubes_actual ≥  SUM[||lighting||.n_set]
⇒    Maintnance_R is confirmed
     & report "spare tubes required in stock but may be allocated to another job"
```

The two rules shown above check the lighting spares records to ensure that sufficient tubes are available to replace the lights in the area. These are included just to give a warning to the operator and not to set a limiting factor on the scheduling. Two rules are included so that in either case a rule is found to be true. If the stock level is too low, a warning is given to re-order the required number. The second rule could be extended to check the diary data-base to see if the parts are allocated to another job.

If the lighting installation is not maintained a cost is incurred in spot replacements of faulty luminaires and light is wasted due to inefficient tubes and dirty light fittings. The cost of this lost light can be estimated. Conducting a bulk replacement too early means that a small amount of useful tube life is wasted. The most economic time to schedule maintenance is just before the predicted cost of not conducting maintenance exceeds the cost of conducting maintenance. This can be calculated for lighting as follows.

The following terms are defined:

C_u = unit cost of tube + starter [£]
C_{lb} = labour cost of bulk planned maintenance including cleaning the lamp [£]
C_{ls} = pot labour cost for maintenance of one luminaires [£]
n = number of lights on a particular floor
W_t = power rating of tube [W]
C_e = light cost [£/kW/h]
P_f = Probability of failure
P_{uh} = Predicted use [hours]
A_{lh} = Average life [hours]
L_l = Light level as a fraction of the maximum

The total cost to conduct bulk maintenance at a particular time *t* is given by,

$$C_m = \sum_{1}^{n} (C_u + C_{lb}) \qquad \ldots\ldots \qquad (17)$$

The total light breakdown cost until time *t* is given by,

$$C_b(t) = \sum_{1}^{n} [P_f(t) \times (C_{ls} + C_u)] \quad \ldots\ldots \qquad (18)$$

The remaining unit cost of fittings is,

$$C_{ur}(t) = \frac{C_u}{A_{lh}} \times \sum_{1}^{n} \left[A_{lh} - \frac{1}{n} \sum_{1}^{n} P_{uh}(t) \right] \quad \ldots\ldots \qquad (19)$$

The light running cost is,

$$C_r(t) = \sum_{1}^{n} \left[\frac{W_t \times C_e}{1000} \times P_{uh}(t) \right] \quad \ldots\ldots \qquad (20)$$

The cost of lost light is therefore,

$$C_l(t) = C_r(t) \times L_l \qquad \ldots\ldots \qquad (21)$$

the cost of maintenance per unit time as a function of the maintenance interval t is therefore,

$$C(t) = \left(\frac{C_m + C_b(t) + C_l(t) - C_{ur}(t)}{t} \right) \quad \ldots\ldots \qquad (22)$$

The economic maintenance time is when the cost to undertake a bulk maintenance equals the sum of the cost of the lost light output and cumulative breakdowns. This is allowing for the cost of the remaining life in the surviving tubes at the maintenance time. This occurs when the total cost is at a minimum. Appendix H gives the results of some analysis using data from the Lloyd's lighting control system. Results show that savings of several thousand pounds (15%) are possible depending on the luminaire usage and minimum acceptable light levels.

Another possible saving is in the reduction in the number of light readings which have to be made. As outlined previously, once the depreciation constants are known for the particular installation and light fittings the light level can be predicted with some accuracy at any future time. This would make use of the zone's light level sensor and the *|BUILDING|.light_level_measured* property of the 'Building' objects. This would reduce the maintenance staff's time in taking light reading measurements resulting in a lower maintenance labour cost.

At the time of a bulk replacement the majority of the luminaires are the originals from the previous bulk replacement. Around 10% or less are newer luminaires installed as the result of spot replacement of failed luminaires. The hours of use for these luminaires will vary from very little to almost as long as the original luminaires where their installation was to replace luminaires that failed very early on in their life possibly as the result of damage in transit. The IBMS has data relating to each individual luminaires and thus those with a predicted useful remaining life may be numbered and used for spot replacements during the next maintenance cycle. The system can advise on exactly which luminaires to use for a particular spot maintenance. In practice, such a system would involve storing about 8% of the tubes following a bulk replacement and the introduction of a numbering system administered by the IBMS. One must also allow for the cost of storing the used lamps, maintenance staff training and the consequences of human error in running such a system.

The rule-base for lighting also includes features for interrogating the mathematical model and answer questions on hypothetical situations. For example, "when will the light level fall to 40%? What is the total cost of conducting a bulk maintenance on lighting in February? What level of spares need to be kept if maintenance is conducted in March?" These are answered by the user setting the constraints and interrogating the mathematical model.

5.5.7 General Approach

Lighting has been used to illustrate this approach, in a full system a class of scheduling rules would be written for all types of plant. Each plant class definition would have its own method definitions for *cost_per_time_predicted* and *failure_probability_predicted.* A mathematical function is derived which represents the cost of maintenance per unit time. This should include the possible waste of energy or materials if maintenance is delayed and the cost of new parts and labour. This function should have a minimum value for some time interval t. A survival probability function is found which best matches the historical breakdown data for this class of equipment. The human factors are then found based on constraints imposed on the operating performance of the equipment.

For example, the same approach can be applied to filter maintenance for air handling units. In time filters become blocked which restricts the air flow. In VAV systems, the fan will compensate for this which increases its power consumption. The cost of increased electricity used by the fan must be equated with the cost of changing the filter whilst it still has some useful life. During the winter, the air handling unit is unlikely to be operating at full capacity and thus the blocked filter may not effect the system's operation at all. The effect of the blocked filter could be predicted by knowing the air handling units predicted use for a particular time of the year. Finally, human and resource factors must be considered; the indoor air quality should not drop below a minimum set level for a particular area. It would save time to change the filter at the same time as the air handling unit is taken out of service for some other maintenance.

5.6 Summary

The work has shown that the use of knowledge-based systems within an IBMS, used in conjunction with a graphical user interface, can provide many enhanced facilities and assist the IBMS operators in their work. The inference functions are illustrated in Fig.5.16.

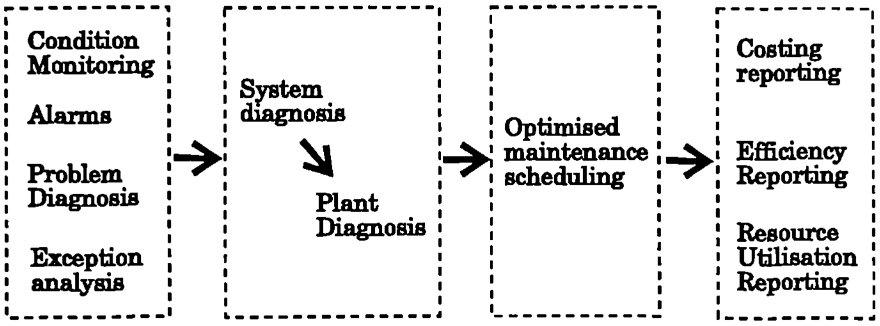


Fig.5.16 Inference Process for Facilities Management

The work has investigated two areas of assisted operator procedures: handling a problem reported over the phone from an occupant or a problem identified by the IBMS itself. In both cases the operator is asked questions and is guided in the best action to take. The IBMS control system identifies problems either through condition monitoring or by comparing the plant's current performance with its past history. The system can predict future breakdown patterns, advise on specific maintenance following a breakdown and help to highlight common and particularly costly faults. The system can then assist in developing a preventative maintenance routine to minimise the failures and down time. The rule-base approach can schedule work at the most economic time by offsetting the predicted cost of plant failure against the cost of preventative maintenance.

To be as efficient long term as an experienced human operator the diagnosis system must also be self-tuning. A methodology has been proposed to incorporate this self-learning characteristic by linking it to historical breakdown data using artificial neural network intelligence. Fault and symptoms are pre-defined for classes of plant equipment. The artificial neural network learns the relationship between this data and any other external influences. Once defined, such a system requires no programming in the traditional sense and would be adaptive to the changes in the building as it is trained on actual breakdown data. It does have the disadvantage that adding more symptoms or faults within a class requires modifying the neural network and retraining whereas rules in a rule-base can be added or removed without necessarily effecting the operation.

The maintenance diagnosis and scheduling rule-bases were tested with rules written for the HVAC and lighting classes of equipment. These were investigated using the same data-base as was used for the software control simulation outlined in chapter four. A prototype system, which includes a graphical user interface developed with Microsoft's ToolBook graphics, was installed in the Lloyd's building for a trial period. Following the evaluation period, favourable feedback was received from the Lloyd's facilities management staff. The Lloyd's facilities manager was particularly delighted that the information was being presented to him in an understandable fashion.

CHAPTER SIX

CONCLUSIONS AND FURTHER WORK

6.1 Overview

The goal of facilities management is to manage the building and its services in the optimum way. To assist is this aim, an 'intelligent' integrated building management system should effectively manages resources in a co-ordinated manner to enhance the well-being and productivity of the occupants. At the same time, it should show cost savings compared with the conventional approach. This is in agreement with the definition of an intelligent building given in chapter one. Current BMS systems do not fully meet these requirements.

This thesis proposes a methodology whereby improvements can be made to the control and management functions within building management systems through integration and the use of artificial intelligence techniques. The work adopts an integrated structure with a central control system linked by a common building data-bus to numerous autonomous outstations around the building. Each outstation is responsible for the full environmental control and security of a specific zone or area. This thesis has proposed two IBMS knowledge-bases which can be incorporated into the IBMS software. The first provides an integrated control approach to all the building's services and the second is concerned with facilities management applications. The control knowledge-base is processed continuously, partly on the central IBMS control system and partly on the numerous outstations. This second knowledge-base assists the IBMS operator in diagnosis problems and faults within the building systems and decision support for preventative maintenance scheduling. The rule examples given in this thesis are not given as a complete set but rather as examples of what is possible given the information and control function capabilities. A disadvantage of many computer management systems is that the reasoning behind their actions can be puzzling. A rule-based approach has the advantage that, by its very nature, it can explain its deductive reasoning to the operator by listing the rules leading to the particular action. This is quite unlike the traditional algorithmic approach. This feature is very desirable to BMS operators, since knowing the reasons behind a suggestion or action enables the operator to have more confidence in the system.

6.2 Conclusions

The two hypotheses stated in chapter one were that a rule-based approach to integration would firstly lead to increased energy savings by tuning the energy consumption to the use of the building whilst causing least inconvenience to the occupants. Secondly it would make the building easier to maintain by incorporating 'expert' knowledge into the management systems. In every case explored, the results are consistent with the hypotheses and point strongly to improvements in the running of high-tech buildings.

Energy savings can be achieved by incorporating a self-tuning occupancy predictor into the zone controller and by using rules to control the environment within the building for maximum occupant comfort at minimum cost. The full energy saving potential is often not obtained from current energy management systems as the users of such systems lack the time required to reset set points and occupancy schedules. The rules enhance the control in the following ways:

- The optimum start and stop cycles are tuned to fit the expected occupancy patterns using a mathematical thermal model of the area. If the model's prediction is incorrect, rules exist to modify the model. It was found by experiment that the thermal model could predict optimal start and stop times to an accuracy of 3% for various external temperatures. The results show a good correlation despite the simplifications involved in the mathematical model. However, these results were obtained during a two week period, during which large variations in external temperature did not occur. Nevertheless, the self tuning aspect should keep the system in tune under varying conditions.
- To prevent the controller from making incorrect decisions on environmental control a combination of actual and predicted occupancy is used. This prevents the controller switching the lights off when someone leaves the room for a brief period or when the area's occupancy suddenly changes from the predicted pattern. In all cases the controller errs on the side of caution.
- It has been predicted by computer simulation that electricity savings of up to 20% would be potentially possible for lighting and heating by tuning the control of building services to the use of the building.
- Rules are included to prevent excessive wear on equipment caused by too frequent switching and to control the environment in areas not continuously occupied by relating their activity to other areas of the building.

- The global control system overrides the occupancy prediction when an area is known to be occupied and, if required, automatically takes plant out of service prior to maintenance or routine inspection.
- Emergency situations, such a load shedding, are handled with least inconvenience to the majority of the occupants using a system of variable load priorities.
- It has been shown that current energy saving schemes such as load control and load balancing can easily be incorporated into the rule-based control methodology.

It has been shown in chapter five that an integrated rule-based approach would make the building easier to manage and maintain by incorporating more on-line 'expert' assistance into the IBMS. Building management systems require more technical expertise to maintain than buildings with more simple control systems and, if not managed correctly, can have higher maintenance costs. This often leads to the facilities management team entering into expensive maintenance contracts with their suppliers or contractors. It was reported by the chief engineer of Lloyds building that maintenance labour whether contract or in-house is the single most expensive cost of running the Lloyds building. The rule-based approach could lead to further cost savings in terms of the maintenance staff's time in the following areas:

- In the event of equipment failure rules exist to identify abnormal operation and signal problems to the operator.
- The IBMS attempts to compensate for equipment failure by utilising reserve equipment where this is made possible by the design constraints.
- The facilities management knowledge-base provides a series of utilities which can be called upon to assist in diagnosing faults in an engineering system or at plant level and in identifying trends and suggesting possible causes. This saves the time of both the skilled BMS operators and maintenance staff.
- Preventative maintenance schedules are tuned for optimum economic benefit without causing undue inconvenience to the building's occupants. This reduced down time and saves money on unnecessary maintenance. Case studies conducted using data from the Lloyd's building have suggested a cost saving of 15% by integrating control data into the planned maintenance schedule.
- For such maintenance features to be as effective as a human operator, a self-learning fault prediction methodology has been proposed using artificial neural networks.

The prototype system has been evaluated by facilities managers at both Lloyd's and Brunel. At Lloyd's the system was installed in the building for a two week trial period using off-line data from the Lloyd's CALMS maintenance system. The control room operators and engineering staff particularly liked the on-line advice and the automatic tuning of optimum preventative maintenance schedules.

An object-oriented data structure has been proposed in which all the key information required by the rules is classified into distinct classes. In practice, object orientation design takes longer to develop than conventional structured language implementations. The advantage lies in the subsequent modification and maintenance. The object orientation is important at the global control level and for the facilities management knowledge-base. Here it would be impracticable to write rule-bases for each application. The rules have been written around classes of equipment and objects which are described in terms of properties. This detaches the knowledge, coded as rules, from the actual data describing the building and its systems. The rules proposed in this thesis are valid whether there is only one zone controller or several hundred. Similarly, within a class of equipment such as air handling units, the rules are valid if a particular air unit has 3 filters of 33 filters. Modification of a particular object can take place without requiring major changes to the knowledge-base. Only the rules and methods relating to the object required changing. For example, if the building's heating system was changed from water-filled radiators to heated air units the method to calculate the failure probability would require altering along with the values of the properties describing the components of the system such as their average life, power rating and unit cost. The rules such as maintenance scheduling, load control, load balancing and related area control would all remain valid. This is an important quality for the commercial viability of such a system, as it would be impracticable to write custom rule-bases for every application.

6.3 Limitations

One of the main regrets with the work was that some areas of it could not be verified by implementation. Initially it was hoped to use the systems in the Lloyd's building as a test bed. However, during the work it became apparent that, for security reasons, the restricted level of access provided to their systems was not sufficient. This necessitated the software simulations. Fortunately Brunel University made available a room to investigate the behaviour of the zone controller. Thus the zone control work was implemented at Brunel, whilst the global control system and the facilities management knowledge-base were investigated using a software simulation, based around data from the Lloyd's systems.

The emphasis of this work has been in the rule-based concepts and not in the actual hardware implementation. Hardware design was pursued no further than was necessary to investigate the rule-base methodology. This has meant that the actual networking of zone controllers was not considered. If such a system were to be fully implemented, one must consider the physical timing problems inherent in networking and message prioritisation. Fire and security threat messages must take priority over routine environmental parameter updates. The proposal outlined in chapter four is for emergency signals to interrupt the global controller whilst all environmental data is obtained by polling the outstations. This requires further investigation to prove its viability. Alternatively, some form of direct memory access could be used by the controllers to update the central database and a token ring network with different levels of message priority may be advisable.

One of the biggest problems in implementing such a system might be in making it fail-safe or at least failure-tolerant. This could be achieved by incorporating a certain degree of built in redundancy. Structuring the network with a ring of controllers builds in redundancy as there are two paths for communicating with the main controller. Also, zone controllers can operate autonomously in the event of central controller failure. A backup central system could be incorporated in a large building allowing one of the zone controllers to run some of the high priority functions of the central control, such as load-shedding and fire control, in emergency situations. Provision for emergency power must be made throughout the system. This would include back up power for the zone controllers using lithium cells and an uninterruptible power supply for the central controller. As proposed, the system is to some extent fault tolerant as it includes fault identification for the zone controllers themselves. Each controller includes the ability to detect and report on problems with their sensors and report abnormal readings. To allow for total zone controller failure, control relays should be wired so that the circuit fails safe. Alternatively, in the event of a zone controller failure its control function could be merged with a neighbouring controller. This could form a part of the reserve equipment utilisation rules and would require all plant to have direct access to the LAN in the event of an outstation failure.

The processing speed of a rule-based system may be of concern. Rule-based systems in general require more processing time than conventional algorithmic approaches. The prototype system used Nexpert for assessing the global control rules. Nexpert is a high level rule-based development system and as such is very slow even on a relatively fast machine such as a 33MHz 486 system. Whilst Nexpert has been ideal as a development tool, for increased speed and flexibility the system would need to be written in a compiled object-oriented language such as C++. The central controller computer must be sufficiently powerful such that it can process the data from the zone controllers and not spend the majority of its time accessing their data. The current steady advances in hardware and processing speed should mean that this will not be a problem with the next generation of micro-computers likely to be around the 150MIPS mark. One must also ensure that the communications network is fast enough. The BMS should take no more than a few seconds to respond to a fire threat transmitted from the furthest outstation. This should include the time taken to acknowledge the signal, raise the alarm, take action to prevent the fire spreading and initiate the evacuation sequence.

Another problem is in defining a network protocol that can handle a large number of nodes while still providing quick response for alarm and emergency signals. In addition to this, each node must be able to interface with the outside world, to control fans, lighting, humidifiers, etc. and to accept a wide range of inputs from sensors. A solution may be about to come through a new product which is about to be launched by Echelon, Motorola and Toshiba called the Neuron chip. Instead of defining a network protocol Echelon have developed a micro controller chip with networking and interfacing built in. The chip consists of three microprocessors multiprocessing and using shared memory as a communication link. Networking is based around a protocol called LonTalk which is used for communicating between Neuron chips over various media including twisted pair, radio or power lines supporting up to 32285 nodes. An enhanced feature over standard Ethernet networks is the ability to prioritise messages which can be used for alarms thus ensuring rapid reception and acknowledgement. Besides providing the communications link, the chip also supports an 11 channels of input/output ports which can be used to control equipment or gather information via sensors. The Neuron chip could take the IBMS hardware structure one stage further by eliminating the outstation in its current form.

The proposed hardware structure advocates more sensors and monitoring points than are available in current BMS. These sensors include light sensors, occupancy sensors and power consumption metering. This would lead to an increase in costs at the commissioning stage or subsequent retrofit. An IBMS incorporating people counting is not a new idea but it has yet to be successfully implemented in modern systems. Possible reasons for the lack of this mode of control are,

- The lack of integration within existing IBMS systems to make use of people counting information
- The fact that, for many applications, people counting is not sufficient - what is required is accurate prediction of occupancy levels within the building
- The high cost of fitting a building out with people counting sensors
- The fact that a sufficiently reliable occupancy sensor has yet to be developed

If the zone controller included security functions, as proposed in chapter two, the final two problems may then be overcome using smart card technology. Smart cards[36] have been available for some years now and are becoming very popular in security applications. The next generation of smart cards will contain miniature radio transmitters that will transmit the data from the card to the sensors thus saving the user from placing the card in a swipe reader. Thus an individual can walk around the building and the security system knows where in the building they are at any time. Although this sounds akin to George Orwells 'big brother' it is already a reality in some buildings in the U.S. where the possibility of leaving ten minutes early on Friday afternoon undetected is a thing of the past!

If the self-learning feature for fault diagnosis and prediction were to be incorporated, it would require software to regularly switch between 'learn' and 'interrogation' modes particularly when adding new symptoms or problems to the system. This may require a rule-base to identify a new fault which should be added to the system whilst recognising an unexpected outcome from an existing classified pattern of symptoms.

The optimised maintenance scheduling relies on statistics for predicting the breakdown probability. Generally, events do not necessarily follow statistics and there is the possibility of errors arising. Different modes of operation may give rise to different failure characteristics and further work is required to prove or disprove this possibility. The rules based on heuristics for the 'human factors' need to be flexible and changeable by the operator. In some circumstances they may need to be dependent on specific areas or events. For example, the minimum allowable light level will depend on the function of the area. This would necessitate further properties, methods or backward chaining to specify such thresholds depending on the situation.

Although more on-line fault diagnosis would be advantageous to the building's operators and management it is possible that IBMS manufacturers would be reluctant to implement it. There seems to be a trend for the manufacturers of high technology equipment, such as building control systems, to reduce the selling cost of a product to a minimum and then to make as much money as possible by selling expensive maintenance contracts. Elementary application of AI technology could allow many electronic systems to conduct self diagnosis, thus reducing maintenance time and cost. However, in many cases, it seems that it is a policy

---

[36] The card is like a credit card but it contains a small microprocessor and a little random access memory. The card when placed in a card reader can identify the owner along with their security clearance. This allows the IBMS to admit people to areas of the building in accordance with their security clearance.

decision to make the error reporting as obscure and unhelpful as possible just so that the customer has to call in the supplier's own maintenance staff to correct the fault[37].

Providing too much information to IBMS operators could have some disadvantages. One must not forget the old adage 'a little knowledge is a dangerous thing'. Care must be taken to ensure that on-line advice does not give someone the false self confidence to attempt to go further than the recommended actions given and in so doing cause more damage than they solve. Also, operators may take too much notice of computer diagnosis and lose the ability to think and to reason for themselves. On-line advice should be used only as a tool, in the same way as an electrician might use a multi-meter.

6.4 Further Work

It is envisaged that the work could be extended in several ways. These extensions fall into two main areas: extending the knowledge-bases to cover more areas within control and management or investigating the design of IBMS.

6.3.1 IBMS Control

The load management feature of the global control system could be extended to make use of local generators for reducing the peak demand. This is an idea which was recently considered by Lloyd's engineering department and is called peak lopping. This could be implemented by writing rules which make the decision based on offsetting the generator's operation and start-up costs against the savings made in electricity and reduced tariffs. This requires knowledge of relative energy costs which can be stored within the resources database. In large buildings, the zone energy control of air handling systems incorporating air cooling, may make use of what is known as a night purge. In the early morning the outside air temperature is usually below the temperature inside the buildings. The night purge function pre-cools the building by filling it with the cooler outside air before the air handling systems starts thus delaying the actual start up time for the cooling systems resulting in a decrease in energy consumption of the cooling systems during start up. Rules could be written to detect when a night purge can be applied based on the occupancy time of the area and the relative temperatures. The control system could make use of weather, in addition to occupancy prediction, to schedule services within the building. For example, knowledge of future outside temperature would be useful

---

[37] The author once worked for a major manufacturer of electronic test equipment. Most of their products incorporate on screen fault diagnostics which enables a relatively lowly qualified technician to correct faults. This feature is a company secret and is not revealed to the customer. The company charges high prices for their maintenance contracts.

for optimum start and optimum boiler temperature setting. Wind speed and direction could also be incorporated into the optimum start and stop rules. The mathematical model should be adapted to take account of thermal losses in areas serviced by a ventilation system as the model only considers a water heating system scenario.

The rule examples given in this thesis only explore the control of lighting, HVAC and some electrical loads. The work can easily be extended to include other areas such as elevators, fire prevention and security. In the case of elevators the knowledge-based system can be used in conjunction with occupancy detection to improve elevator algorithms. If an elevator system were to be aware of the number of people waiting and their desired destinations, the rules could determine the optimum car movement to transport people more efficiently and reduce waiting times [90]. In the case of fire systems the global control rules can be written to generate audible and visual alarms as well as measures to reduce the spread of the fire and evacuate the people. This includes activating appropriate fire extinguishing systems, operating smoke vents, closing air conditioning fans in affected areas, unlocking security doors, pressurising and lighting escape routes, recalling lifts to ground level, disconnecting selected electrical supplies and reporting the occupancy status to fire crews. In the case of security system, the occupancy data can be used in conjunction with the 'DIARY' class data to provide security access clearance for 'out of hours' occupation. An important secondary benefit is the calculation of time attendance hours for input to the payroll computer. The ability to audit personnel movements is also a key factor in the event of fraud or other misdemeanour.

6.3.2 Facilities Management within IBMS

The work can be extended in many ways both in terms of the GUI environment for operators and the assisted operator procedures. A great deal of work has been done in the field of user interfaces and the work moves into the field of psychological sensors and human behaviour. Besides this the user interface's presentation manger could easily be linked with the diagnostic system to automatically switch the graphical display to the required menu in emergency and fault handling situations. This would save time in navigating through menus. A further requirement is for the GUI's objects display to automatically configure itself to the plant database. Some systems, such as the current TREND system, allow the user to manually edit the graphical displays. A display linked to the plant database would require less commissioning and subsequent system modification time when the plant database is changed.

Lighting maintenance has been a useful area to investigate for optimised preventative maintenance scheduling as the information was easily available. However, it must be realised that significant cost or time savings would not occur by implementing these ideas alone. Optimised lighting maintenance must be seen as only a small part of the whole integrated maintenance management system. To make significant cost savings these ideas must be applied to the areas of maintenance that currently have the highest expenditure. In many buildings this is likely to be the HVAC systems.

The fault prediction could be extended to include system failure probability theory. For example, the failure of a particular system is a function of the individual failure probabilities of its individual sub-systems. This could be represented in the 'PLANT' class and sub-class structure as defined in chapter three. There is also the possibility at a later date of integrating visual maintenance repair data, such as drawings and manuals, along with the fault description data and diagnosis information by the use of scanned documents and optical character recognition indexing.

For a diagnosis and on-line advice system to be as efficient as a human operator it must be adaptive and 'learn' changes in the building systems and their operation. The neural network fault diagnosis and prediction work can be extended by including actual run time data. It could be that the use of a neural network within a deep knowledge system is the way forward for self-tuning artificial intelligence diagnostic systems. This could be investigated if more training data could be made available.

If a deep knowledge system were to be built, incorporating a general model of a building, it could be used by the facilities management staff for asking 'what if questions' such as setting constraints and conditions or to find the predicted effect of changing equipment operation. For example, to consider if it would be more economic to reduce or increase staffing levels, re-distribute staffing levels between different disciplines or to reduce or increase planned preventative maintenance. It could predict the consequences of a lift going out of service in terms of increased waiting times for the other passengers. It would be useful for planning events in the building and providing answers to questions like 'Could we hold a conference for 500 people on gallery 7 next week?', 'Would the lifts handle this extra loading without excessive waiting times?'. It could provide information for budgeting and contingency planning such as 'What is the predicted HVAC running costs for next year?', 'If one of the main boilers were to fail, have we still got sufficient heating capacity for 3 hours if it occurred this month or next January?' or 'Can we reduce the lighting costs by a more optimum switching pattern to make more use of daylight?'. Building the knowledge and data required to 'reason' and answer such questions into the BMS would be a further step in adding intelligence to integrated building management systems.

6.3.3 IBMS Design

A logical progression from using knowledge-based systems in IBMS control and management is to use such technology in the design of the building services themselves. There are currently many sophisticated computer aided design packages for use by consulting engineers. As almost all building design work is computer based, it should be possible to use the data from an integrated building services CAD package for the construction of the building management system database. This would reduce the commissioning time. In addition to this, the building services CAD packages could be combined with a KBS which can validate designs not only for structural integrity but also for compliance with building regulations [91]. Furthermore, a deep knowledge system based around this application could be used for finding the optimum design for a particular area by modelling the buildings environmental characteristics. Information from finished buildings could then be fed back into such a model to improve its performance.

Rules could be written for different disciplines such as structural, mechanical and electrical engineering. Such a system would require structuring in such a way that individual rule-bases can be upgraded without affecting the whole knowledge-base when new building regulations are published. This would require writing rules for specific regulation classes in the same way as rules have been written around classes of equipment in the IBMS control and management knowledge-bases. A further stage could be to use constraint modelling techniques for finding the optimum design for building services.

6.5 The Future of IBMS

Even before the world recession began, the UK was behind in developments in high technology buildings. Intelligent buildings in Japan are now years ahead of the UK and the Japanese Intelligent Building market is booming. As far back as 1986 the Japanese government provided substantial tax incentives to developers of intelligent buildings and laid down standards to be used in the planning of such buildings. Japan even has a national Building Research Institute which employs 118 research staff and 54 support personnel to undertake research into building science. To what extent Japan may in the future influence the growth of IBMS in other countries remains to be seen. It is interesting to note that in Japan it is the computer and communications companies, such as Fujitsu, NEC and Toshiba, that are leading the field in IBMS and not the building services industries.

In Europe Integrated Building Management Systems have yet to become clearly defined and widely implemented. Progress by manufacturers has been slow as they have been held up by legislative restrictions, prejudices and resistance to change. Harmonisation of product design in sensing and control across the full range of service functions is slowly being adopted by some of the major companies. Looking further into the future, the common communications standard will enable compatibility to be achieved between the different manufacturers' products which will further extend the benefits and transform the whole of the controls business. When this fully integrated and open architecture arrives it will provide scope for intelligent control and management functions as more information will be readily available to the IBMS control system. The question is: will the IBMS of the future meet the requirements of the facilities management staff and overcome some of the high-technology problems caused by current building management systems?

In the past where high technology systems have been used, they have caused problems for the facilities management staff of high technology buildings. If the way forward is for more integration and high technology systems, as it indeed seems to be, the manufacturers must take more account of the requirements of the people who are asked to use them. The 1980's seemed to be the age when computers and high technology entered most peoples lives and many new applications were found for microprocessor control systems. Unfortunately, in many cases, the users of such equipment do not seem able to access their full potential. A classic example of this is the programming of the home video recorder. Perhaps, through the strategic use of artificial intelligence techniques, the 1990's can become the age when interactions with such systems become easier for the non computer experts who have to use them.

APPENDIX A - CLASS/PROPERTY DEFINITION

This appendix contains a list of class, sub-class and property definitions. The five main data structures are included namely plant equipment, building description, maintenance, resources and planned events. Many properties are used for more than one class and in the system are only defined once, however for clarity their definition and a brief description of their function are given for each class definition. When a property is inherited from a parent class this is indicated using *italics* in the property description.

Section A2 gives the class, object and property definitions for the rule hypotheses, all of which are defined as objects within Nexpert.

A1 Data Classes

A1.1 Plant Equipment Data

Class Name : PLANT

Sub Classes : Elec, Lighting, Hvac, Transport, Security, Fire

Properties :

power_measured	FLOAT	The power currently being consumed by the plant (kW)
class_set	STRING	The name of the sub-class of equipment the plant belongs
location_influence_no_set	INTEGER	The area of the building influenced by the plant's operation
location_no_set	INTEGER	The physical location number of the pant
no_set	INTEGER	The plant identification number
type_set	INTEGER	The particular type, if appropriate, of sub-class equipment
function_set	STRING	A text description of the function of the plant
control_status	INTEGER	Control mode status
reason_actual	STRING	Text description for operator on the load's ranking order
last_maintenance_act	DATE	The last time preventative maintenance was conducted
last_breakdown_act	DATE	The last time this plant experienced a breakdown
scheduled_period_act	DATE	The scheduled preventative maintenance period (week no)
power_predicted	FLOAT	The power consumed by the plant when last running (kW)
cost_hour_pred	FLOAT	The predicted running cost per hour
failure_probability_pred	FLOAT	The predicted failure probability (based on historical data)
maintenance_cost_pred	FLOAT	The predicted cost of preventative maintenance work
use_weekly_pred	INTEGER	The predicted weekly usage for this plant

Class Name : Hvac

Sub Classes : Fans, Pumps, Chillers, Air_units, Boilers, Pressure_units, filters

Properties :

AHU	INTEGER:	Air handling unit association number

Class Name : Fans

Additional Properties :

speed_set`	INTEGER	Fan operating speed in r.p.m.
diameter_set	FLOAT	Fan diameter in (cm)

[similarly for Pumps, Chillers, Air_units, Boilers, Pressure_units, filters]

Class Name : Lighting

Properties :

id_actual	FLOAT	Initial depreciation due to dirt
it_actual	FLOAT	Initial tube light output depreciation
dd_actual	FLOAT	Monthly light level reduction due to dirt deposits
dt_actual	FLOAT	Tube light output depreciation per 1000 hours use
Group_set	INTEGER	Switching group allocation number

Class Name : Electrical

Properties :

importance_set	STRING	Pre-defined priority rating for electrical loads

[similarly for Transport, Security, Fire]

A1.2 Building Description Data

Class Name : BUILDING

Sub Classes : (none)

Properties :

occupancy_measured	INTEGER	The measured occupancy in the area
temperature_measured	FLOAT	The measured temperature in the area (degree C)
light_level_measured	INTEGER	the measured light level in the area (100% of design max)
fire_threat_measured	BOOLEAN	Signal from the fire sensors[38]
temperature_set	FLOAT	The set temperature in the area (degree C)

---

[38] If a rule-base were to be included to interpret the readings from fire and security sensors as proposed in chapter four this property would then bevome a 'confidance factor' and be a FLOAT data type.

light_level_set	INTEGER	the set light level in the area (100% of design max)
location_no_set	INTEGER	Defined location number within the building
AHU_set	INTEGER	Air handling unit associated with this area[39]
description_set	STRING	A description of the function of the area
size_set	FLOAT	The area of the room in square meters
floor_set	INTEGER	The floor number (in multi-floor building)
a1_act	FLOAT	parameter representing the room's thermal response
a2_act	FLOAT	parameter representing the room's thermal response
b1_act	FLOAT	parameter representing the heating system's thermal response
b2_act	FLOAT	parameter representing the heating system's thermal response
clearence_actual	STRING	A flag to indicate security clearance given to an area
activity_actual	STRING	Perceived activity level in the room based on prediction
lighting_off_time_actual	FLOAT	Time when lighting for this zone was last switched off
activity_predicted	STRING	Perceived future activity level in the room based on prediction
Q_Heating_predicted	FLOAT	Predicted heat energy from heating system in area
Q_sundry_predicted	FLOAT	Predicted sundry energy gain in area (time dependent)

A1.3 Maintenance Data

Class Name : BREAKDOWN

Sub Classes : (none)

Properties :

class_set	STRING	The name of the sub-class of equipment the plant belongs to
fault_set	INTEGER	The breakdown fault code number
hours_set	FLOAT	The time required to correct the fault (in hours)
no_set	INTEGER	The plant number which the fault occurred on
period_set	DATE	The maintenance period the breakdown occurred in
trade_set	STRING	The class of repair personnel who conducted the work
date_set	DATE	The date the fault occurred on
n_set	INTEGER	The number of occurrences for this fault

---

[39] In the case of a wet heating system this value would be the heating circuit number.

Class Name : FAULTS

Sub Classes : (none)

Properties :

class_set	STRING	The name of the sub-class of equipment the plant belongs to
description_set	STRING	Description of the fault and its possible causes
hours_act	FLOAT	The average time required to correct the fault
trade_act	STRING	The class of engineer most suitable to correct the fault
likelyhood_pred	FLOAT	The probability of occurring per period
materials_act	STRING	Any special materials or tools required for the work

A1.4 Resources Data

A1.4.1 Parent Class

Class Name : RESOURCES

Sub Classes : Trade, Energy, Stores

Properties :

unit_cost_set	FLOAT	The cost per kWh
resource_status	STRING	Flag to indicate possible unavailability of resource
supply_capacity_set	FLOAT	total hours worked per week
class_set	STRING	Trade classification description
description_set	STRING	Specific resource description
used_period_set	FLOAT	Total consumption/hours worked/used last period

A1.4.2 Sub-classes

Sub-class Name : Trade

Sub Classes : Mechanical, General, Electrician

Properties :

unit_cost_set	*FLOAT*	*The labour cost per hour*
supply_capacity_set	*FLOAT*	*total hours worked per week*
class_set	*STRING*	*Trade classification description*
description_set	*STRING*	*Personnel's name*
resource_status	*STRING*	*Flag to indicate long term illness or absence*
used_period_set	*FLOAT*	*Total hours worked last period*

Sub-class Name : Energy

OBJECTS : Electrical, Oil, Gas

Properties :

unit_cost_set	*FLOAT*	*The cost per kWh*
supply_capacity_set	*FLOAT*	*Total supply capacity in kWh*
resource_status	*STRING*	*The supply security status*
class_set	*STRING*	*Energy classification description*
description_set	*STRING*	*Energy source description*
used_period_set	*FLOAT*	*Total consumption last period*

The Electrical class includes the FLOAT property *power_required_actual* which is used by the load shedding rules.

Sub-class Name : Stores

Sub Classes : Mechanical, Electrical, Special

Properties :

unit_cost_set	*FLOAT*	*Unit cost of item*
supply_capacity_set	*FLOAT*	*Total number of this unit in stock*
resource_status	*STRING*	*Flag to indicate that this part is no longer available*
class_set	*STRING*	Stock allocated to plant class
description_set	*STRING*	*Part classification description*
used_period_set	*FLOAT*	*Total number of these parts used during the last period*

A1.5 Planned Events Data

A1.5.1 Parent Class

Class Name : EVENTS

Sub Classes : Maintenance, Diary, Security

Properties :

event_set	INTEGER	Planned event number
start_time_set	TIME	The planned event's start time
start_time_set	TIME	The event's planned finish date and time
description_set	STRING	Text description of the events reason or purpose

A1.5.2 Sub-classes

Sub-class Name : Maintenance

Properties :

event_set	*INTEGER*	*Planned event number*
start_time_set	*TIME*	*The planned event's start time*
start_time_set	*TIME*	*The event's planned finish date and time*
description_set	*STRING*	*Text description of the events reason or purpose*
class_set	STRING	Plant class for maintenance work
no_set	INTEGER	Plant number for maintenance work

Sub-class Name : Diary

Properties :

event_set	*INTEGER*	*Planned event number*
start_time_set	*TIME*	*The planned event's start time*
start_time_set	*TIME*	*The event's planned finish date and time*
description_set	*STRING*	*Text description of the events reason or purpose*

Sub-class Name : Security

Properties :

event_set	*INTEGER*	*Planned event number*
start_time_set	*TIME*	*The planned event's start time*
start_time_set	*TIME*	*The event's planned finish date and time*
description_set	*STRING*	*Text description of the events reason or purpose*
authority_set	STRING	The person who gave the authority

A2 Hypothesis Classes

A2.1 Control Knowledge-Base

A2.1.1 Zone Control Rules

Zone_control	BOOLEAN	Rules to control zone, in ON - OFF - STOP - START modes

A2.1.2 Zone Occupancy Filter

Dormant	BOOLEAN	Rules to define areas as 'dormant'
Quiet	BOOLEAN	Rules to define areas as 'quiet'
Normal	BOOLEAN	Rules to define areas as 'normal'

Active	BOOLEAN	Rules to define areas as 'active'
Becoming_dormant	BOOLEAN	Rules to define areas as 'becoming dormant'
No_change	BOOLEAN	Rules to define areas as 'no change'
Becoming_occupied	BOOLEAN	Rules to define areas as 'becoming occupied'

A2.1.3 Zone Self-Tuning

Ch_in_ext_temp	BOOLEAN	Detect error due to change in external temperature
Ch_in_time_const	BOOLEAN	Detect error due to building's time constant

A2.1.4 Global Override

Override_start	BOOLEAN	Rule to override area control for planned events
Related_area	BOOLEAN	Rule to override area control for related area
*PLANT*_maintenance	BOOLEAN	Rule to override area control for *plant* maintenance

A2.1.5 Global Energy Saving

Economy_control	BOOLEAN	Rules to reduce energy in dormant or unoccupied areas
Normal_control	BOOLEAN	Rule to return area control to normal
*Plant*_load_control	BOOLEAN	Load control rule for *plant* type
*Plant*_load_balance	BOOLEAN	Load balance rule for *plant* type
for example,		
Ahu_load_balance	BOOLEAN	Load balance rule for *plant* type

A2.1.6 Emergency Handling

Lighting_ranking	BOOLEAN	Plant ranking rules for lighting
Elec_ranking	BOOLEAN	Plant ranking rules for electrical equipment
Ahu_ranking	BOOLEAN	Plant ranking rules for air handling units
in general,		
*PLANT*_ranking	BOOLEAN	Plant ranking rules for *plant*

Load_shed	BOOLEAN	Rule for priority load shedding
Change_ranks	BOOLEAN	Rule for changing ranks
Engage_loads	BOOLEAN	Rule for engaging ranks in priority order

A2.1.7 Condition Monitoring

Cond_mon_*SYSTEM*_problem	BOOLEAN	Rules to identify problems with system
e.g.		
Cond_mon_lighting_problem	BOOLEAN	Rules to identify problems with lighting
Breakdown_comp_*PROBLEM*	BOOLEAN	Rules to identify compensate for *problems*

A2.2 Facilities Management Knowledge-Base

A2.2.1 Fault Diagnosis

FaultX	BOOLEAN	Diagnosis rules for fault X
Find_location	BOOLEAN	Find the location of plant X

A2.2.2 Trend Analysis

Class Name : Trend_S   Trend Search rules

OBJECTS : Trend_Sc, Trend_Scsf, Trend_Scsl, Trend_Scslst, Trend_Scst, Trend_Sfsc, Trend_Sfscsl, Trend_Sfsl, Trend_Sfst, Trend_Sl, Trend_Slst, Trend_Sp, Trend_Spsf, Trend_Spsfst

Properties :

class_set, fault_set, hours_set, n_set, no_set, Period_set, thours_set

Class Name : Trend_ABCD   Trend identification and advice rules

OBJECTS : A1, A10, A12, A1a, A1f, A2, A2a, A2f, A3, A3f, A4, A4f, A5, A6, A7, A8, A8a
B1, B2, B3, B4, B5, B6, B7, C10, C21, C38, C39, C4, C5, C6, C64, C9, D1, D1a, D2, D2a, D3

Property : BOOLEAN

A2.2.3 System Problems

Filter_problem	BOOLEAN	Diagnose problem as due to filters
Humid_problem	BOOLEAN	Diagnose problem as due to filters
Boiler_problem	BOOLEAN	Diagnose problem as due to filters
in general,		
*PLANT*_problem	BOOLEAN	Diagnose problem as due to plant type

A2.2.4 Maintenance Scheduling

Maintenance_h	BOOLEAN	Human factors maintenance rules
Maintenance_r	BOOLEAN	Resource factors maintenance rules
Maintenance_e	BOOLEAN	Economic factors maintenance rules
Schedule_now	BOOLEAN	Optimum scheduling rule
Schedule_earlier	BOOLEAN	Compare factors for scheduling earlier
Scheduler_later	BOOLEAN	Compare factors for scheduling later

APPENDIX B - NEXPERT OVERVIEW

B1 Summary

NEXPERT OBJECT is a very flexible knowledge-based system shell which has been used to prototype the KBS applications investigated in this work. This appendix gives an overview of the Nexpert features used in this application to enable the reader to better understand the structure of the knowledge-bases implemented in Nexpert. For a more detailed insight the reader is referred to the sixteen volumes of manuals and documentation which accompany the software.

Nexpert supports a very wide range of knowledge representation features in particular object orientation and inheritance. The domain is modelled in terms of objects, classes, and properties and the specific properties of objects and classes are called slots. Slots are used to store all of the information NEXPERT gathers from the world. Slots have meta-slots which describe the slots' behaviour. The meta-slots range from simple characteristics such as how to prompt for the value to a detailed list of where to find a value for the slot (retrieve from a database, execute an external routine, ask the user, etc.). Properties can be inherited from a class or object to another class or object. Values can also be inherited from a class or object to another class or object. Certain meta-slots can be inherited from a class or object to another object. Inheritance allows efficiency, as the particular attribute only needs to be declared in one place, it provides consistency as everything which inherits an attribute behaves in the same way, and it provides generality. NEXPERT also supports multiple inheritance. Properties, values, and some of the meta-slots can be inherited down or up the object network.

NEXPERT supports rules which contain all of the domain knowledge. Rules manipulate the slots as well as the object and class structures. Pattern matching and interpretations allows the user to reference objects which are determined at runtime. Thus one can write generic rules which reason on a set of objects which are determined when the rule is processed. One can create dynamic objects and dynamically modify the relationships between objects and classes, thus allowing objects and classes to inherit from different parents at different times. These dynamic objects allows one to model a world whose exact structure isn't known at the time of writing the knowledge-based system. One can then create dynamic links between objects or classes and other objects or classes to reflect changing relationships during processing.

B2 Data Structures Within Nexpert

An object is the smallest item of information in the knowledge-based system. It represents any person, place, thing, or idea in the domain for this particular application. One can describe an application's world in terms of various objects. For instance, in a building management system application, each particular area within a building can be an object, as well as all the components needed to service the building.

A class is merely a grouping or "generalisation" of a set of objects. Objects are specific members or "instantiations" of a class. For example all the lift within a building may fall into the class of transport. Objects may belong to several classes, such as "a fan" is both a member of the class "air handling units" as well as the more general class "HVAC equipment:" Classes may also have many objects, there is the possibility for many to many different relationships. A class can also have subclasses. A subclass is a class which represents a subset or "specialisation" of another class. It is a class in its own right and has all the characteristics of other classes. For instance, "HVAC equipment" could be one class with "Pumps", "Fans", and "Boilers" as subclasses.

Properties describe both objects and classes. Properties have a particular data type: they can be *string, integer, float, Boolean, date, or time.* They can also be multi-valued. Some example properties are: weight, colour, value, time, etc. One can use any number of properties to describe an object or class. While objects and classes may have specific properties, these properties are not limited to any one object or class. Thus other objects and classes can have the same property. Furthermore, since the property is independent of the object or class, it will always have the same data type throughout the knowledge-base. There is one exception to the way properties behave. The special property "value" can have (and usually does have) different data types when it is attached to different objects or classes. Thus the "value" property of the "boiler 1" object may have *afloat* data type, while the "value" property of the "fan 45" object may have a *date* value type, and other objects using the value property can have any other data type.

Slots are used to store property values for objects and classes. Thus they hold all of the information in the application. Any information which comes into the knowledge-based application, whether it comes from a database, from the user, from any external program, or is generated internally, is stored in slots. Slots are properties which are attached to objects. The simplest type of slot, which uses the "value" property, is generated automatically by NEXPERT.

In text, NEXPERT represents a slot as the object or class name, then a period, and finally the property name. Thus a slot representing the power rating of a lamp can be represented in NEXPERT as "lamp_no1.value" or "lamp_no1.power". An object or class can have any number of slots. The number of slots an object or class has is precisely equal to the number of properties the object or class has. A slot, no matter what data type, initially has the value UNKNOWN. This means that NEXPERT has not tried to determine a value for the slot. When NEXPERT tries to determine the value for a slot, one of two things can happen:

- NEXPERT finds a value and the slot takes whatever value has been determined
- NEXPERT does not find a value and the slot takes the value NOTKNOWN

B3 Meta-Slots

Meta-slots describe all aspects of the behaviour of slots. These behavioural characteristics span a broad range including:

- How the system should ask the user for their value when it is needed.
- What types of inheritance strategies to use (breadth vs. depth-first, class vs. object-first).
- What type of inference and inheritance priority it should have.
- Where to find the value (Order of Sources).
- What to do if the value changes (If Change).

Order of Sources and If Change meta-slots are often referred to together as methods since both of them provide lists of commands to execute. In addition, If Change meta-slots are also referred to as demons. Order of Sources and If Change meta-slots can be inherited. Inheritance allows one to define methods at the class level and have subclasses or objects use them or define them at the object level and have sub-objects use them. As with other forms of inheritance, this capability provides both consistency and a utility of expression .

NEXPERT supports multiple inheritance (inheritance is explained in the inheritance section). This means that the target of an inheritance event (a slot) may have several different sources from which to inherit a value or method. If there are two or more parents (or children) which are at the same level in the inheritance space from which a particular slot can inherit, then the inheritance priority will determine which parent (or child) is used. The inheritance priority can be set by putting a value of even a another slot into the inference priority slot within the meta-slot. This is also a useful feature to control the priority of knowledge processing. Slots with higher inheritance priorities are inherited from before slots with lower priorities. By default, all slots have an inheritance priority of 1.

In addition to setting inheritance priorities which determine how the slot will compete with other slots when children or parents want to inherit from it, the inheritability meta-slot also determines whether or not a slot can be inherited from at all. There are global inheritance defaults which are explained in the Inheritance section. These defaults determine what can be inherited for the vast majority of the slots. However, some slots may display a behaviour which is peculiar to only that one particular slot. This behaviour can be set in those particular slots' inheritability meta-slots. The inheritance strategy meta-slot determines the breadth-first depth-first and class-first object-first types of conflict resolution. The default inheritance strategy is breadth-first, class-first.

The Order of Sources meta-slot determines where a slot will get a value when SOURCES it is needed. The Order of Sources contains a list of actions to perform to determine the value of the slot. These actions are performed in order, from top to bottom. If a value is found at any point in the list, then the rest of the sources are disregarded. When NEXPERT needs the value of a slot, it will perform the following series of actions.

(1) Check the Order of Sources (OS) defined for the slot.
(2) If any OS is written for the slot, then the system executes the actions in the OS sequentially, from top to bottom, until a value is determined. if the slot's own OS list fails to find a value, skip to step 4.
(3) If the slot has no OS written, then check the parent's OS. If the parent *object.property* or *class.property* has an OS written, the slot will inherit the actions and execute them as its own. If the slot's inherited OS list falls to find a value, skip to step 4.
(4) If the slot's parent has no OS written, then use the default OS strategy. If the slot is a hypothesis use backward chaining to evaluate the hypothesis. Inherit the value *down* from a parent or inherit the value up from a child. If the value of the slot is still not found go to step 4.
(5) If the slot's value is still not determined after completing steps 1, 2, or 3, then the system prompts the user to enter the slot's value.

When NEXPERT needs the value of a slot which doesn't have a value (i.e. it is UNKNOWN), it will use an Order of Sources, which is a list of different possible sources, to try to get a value. It first tries to use an OS declared locally then one which can be inherited, and finally the default OS as follows:

- Backward chain
- Inherit down
- Inherit up.

NEXPERT executes one of these three lists of sources. If the list of sources, whether they be defined, inherited, or the default, all fail, then NEXPERT will prompt the user for the value of the slot. these are only the default Order of Sources. One can define any other sources required in any order but as NEXPERT determines a value for the slot, NEXPERT will exit the method and the rest of the sources will be disregarded.

- *Initvalue* (this initialises the slot to a particular value at the start of a session. Note that this Source is unlike any of the other Sources in that it is executed at the start of a session rather than when NEXPERT needs a value)
- *RunTimeValue* (the slot takes this value when NEXPERT needs it to be determined thus the slot will remain UNKNOWN until NEXPERT needs the value. This serves as a default value)
- Retrieve will retrieve data from a database

Order of Sources can be inherited down from parent objects or classes. This means that one can define a series of sources at the parent level, and all of the children can take advantage of the sources. This topic will be fully dealt with in the Inheritance section.

- The If Change meta-slot lists a series of actions to perform after the value of the slot is changed. There are several very important points about If Change actions:
- If Change actions are performed *immediately* after the value changes.
- When a series of If Change actions are defined, all of the actions are executed from top to bottom after the slot's value is changed.
- By default, there are *no* If Change actions. Contrast this with the Order of Sources meta-slot which has a series of default sources.
- Similar to Order of Sources, If Change Actions can be inherited down the object network.
- The If Change meta-slots will not be executed when a particular slot is reset to UNKNOWN using the Reset operator.

B4 Rules

NEXPERT's capability of providing an intuitive way to represent our domain is a tremendous asset, but we also need to have some way of reasoning on it. Rules provide the ability to reason on the objects within the domain. Rules capture the knowledge necessary to solve particular domain problems. Rules represent, among other things: relations, heuristics, procedural knowledge, and the temporal structure of knowledge. Rules have three basic parts:

- Left-hand side (LHS) conditions
- The hypothesis which is a Boolean slot
- The right-hand side (RHS) actions.

Conditions, rules, and hypotheses are all Boolean data structures. Similar to Boolean slots, they may have one of four basic values: UNKNOWN, TRUE, FALSE, or NOTKNOWN. The conditions represent a series of tests to determine whether or not the hypothesis is TRUE. If *all* of the conditions are TRUE, then the hypothesis is set to TRUE and the right-hand side actions are all executed.

A rule's value depends on the state of its LHS conditions:

- If no attempt has been made to evaluate the LHS conditions, then the rule will be UNKNOWN
- If NEXPERT evaluates all of the LHS conditions to TRUE, then the rule is set to TRUE as well
- If NEXPERT has tried to evaluate the LHS conditions, but could not determine the value of at least one condition, then the rule will be set to NOTKNOWN
- If NEXPERT evaluates the LHS conditions and one of them is FALSE, then the rule will be set to FALSE as well.

NEXPERT rules are symmetric, i.e. they have no inherent "direction" in them. This means that the rule can either be processed in the forward direction by forward chaining events or in the backward direction by backward chaining events. As the rules are symmetric so there is no need to write one set of forward chaining rules and another set of backward chaining rules.

All slots used explicitly in the LHS conditions or the RHS actions of a rule are called data. A hypothesis, in and of itself is not a datum, but if it is used in the LHS conditions of another rule, then it is a datum as well as a hypothesis. Hypotheses which are also data are referred to as sub-goals. It is also possible to manipulate slots which are not data by means of interpretations or pattern matching.

When NEXPERT evaluates the conditions in a rule, it evaluates the condition which has the data with the highest inference priority first. By default, all data have inference priorities of 1, and when all the priorities are equal, conditions will be evaluated from the top to the bottom.

All rules have one and only one hypothesis. However, a hypothesis can have many different rules leading to it. The hypothesis is a Boolean slot. If all the conditions on the left-hand side are evaluated to TRUE, then the hypothesis is set to TRUE as well.

The right-hand side actions are only executed if the rule is evaluated to TRUE. In contrast to the other two parts of a rule, RHS actions are not required. They are a series of consequences of the rule being fired which are executed as soon as the rule is verified. There may be any number of RHS actions.

B5 Inheritance

These representation mechanisms are quite useful in terms of structuring a world, but inheritance is what gives the greatest utility to this form of representation. There are three fundamental types of inheritance:

- Property inheritance
- Value inheritance
- Meta-slot inheritance.

Property inheritance refers to the ability for an object to inherit the existence of a particular property from a class (or a subclass from a parent class). This means that an object, such as "B", which belongs to a class "Patients" that has the property "temperature", will also inherit that property: Property inheritance occurs *immediately*. This means that as soon as an object is added to a class or a property is added to a class, inheritance occurs before anything else.

The second type of inheritance, value inheritance, is the ability for a slot to assume the value of one of its parents (or children) if its own value is UNKNOWN. For example, there may be a situation where NEXPERT doesn't know the maintenance interval for the object "fan 14" is in, but it does know the class "FANS" has a maintenance interval of 26 weeks so "fan 14" inherits this value.

The important difference between value inheritance and property inheritance, in addition to the fact that a value is being inherited instead of a property, is that the value is only inherited when NEXPERT needs the value. If NEXPERT doesn't need the value, it won't needlessly propagate values around the object network.

Once again, this leads to a great utility in terms of expression. Instead of specifying each of the individual slot values, the generic value can be specified at the class level and inherited by any of its objects when the value is needed. The third type of inheritance is inheritance of meta-slots. Meta-slots are behavioural characteristics of slots. General behaviours can be specified at the parent object or class level, and they will be inherited when they are needed.

The default strategy also allows properties to be inherited down from classes to subclasses and from classes to objects, but not from objects to sub-objects. Thus if there is a class "HVAC" with a subclass "Boilers" and a property "cost" is linked to the class "HVAC," the property will immediately propagate down:

Both Order of Sources and If Change meta-slots can be inherited in a manner similar to properties and slot values. Also analogous to slot values, they are inherited only when needed and there are no meta-slots at the current level. Order of Sources and If Change methods are never inherited up. Inheritance of meta-slots proceeds along the same links as for inheritance of properties and values, except for the fact that they can only be inherited down, regardless of the current strategy.

- When NEXPERT needs the value of a slot:
- It looks to see if the slot has anything in its Order of Sources meta-slot. If so, these are executed.
- If not, then NEXPERT tries to inherit an Order of Sources down from one of its parents.
- If none of them have anything declared (or if inheritance down is disabled), then NEXPERT resorts to the default Order of Sources: backward, InheritValueDown, InheritValueUp.
- If all of these Order of Sources fail, NEXPERT always asks a question.

The inheritance as described is the default strategy but NEXPERT uses when new properties are added to classes and objects, new values are assigned to slots, and NEXPERT needs the values of particular slots. However, NEXPERT allows one to completely customise how inheritance takes place.

Preventing inheritance down from classes is useful in the type of situation where classes are generalisations of objects but they don't share the same properties and thus there's no reason to needlessly propagate properties to all of the objects.

A particular object or class may often have several parents or even an entire network of possible parents from which to inherit. Each time an inheritance event occurs, there is the possibility for conflict between alternate sources of information. One specifies how to search for a value using general search strategies. The strategies are applied each time there is a conflict during an inheritance event.

B6 Dynamic Structures

When the exact objects and their relationships are not known at the time of specifying the system of when a knowledge-based system is to be generalised and not specific to one set of objects dynamic structures are used. NEXPERT allows applications to create dynamic objects and dynamic links. The objects and links are created at runtime rather than being compiled with the rest of the application. As the system realises the need for new objects and new relationships, it can create them. This means that one only needs to hard code those objects and relationships which are always used, while the objects and relationships whose existence depends on the current state of the system and the external environment can be created as needed. This also saves both memory and disk space as only the permanent objects and relationships which are needed are created and stored in memory.

Dynamic objects are created at runtime rather than being compiled with the rest of the application. They are deleted when a new session is started. Since these objects are defined during the processing of the knowledge-base, the only way they can have properties and meta-slots is by linking them to classes and relying, on Inheritance Mechanisms. A NEXPERT application can also create new links between both complied objects and classes as well as dynamic objects. Newly created objects can be linked to other objects or classes (whether the other objects and classes are dynamic or not). This linking can be done as soon as the object is created or at any time during the inference process. Links can also be created between complied objects and other objects or classes. Links can be deleted from any objects or classes. Thus the whole object network can be altered dynamically while NEXPERT is running new objects can he added to the network, objects can be made sub-objects of other objects or instantiations of classes, and the links can all be destroyed.

Often the exact objects or classes whose slots one wishes to test in the conditions of a rule or send to a particular routine or function are not known before the inferencing session. In this situation, it is necessary to generate the objects or classes at runtime rather than explicitly naming them in the rules. NEXPERT allows one to use interpretations to implement this strategy. An interpretation is a slot value which is interpreted to be the name of an object or class. Interpretations can also be used in the RHS actions. In this case, the exact same behaviour we have described for the LHS conditions applies. In addition, interpretations can be used within the arguments (third column of the conditions) to the database Retrieve or Write statements. They can be used in the Begin statement to determine dynamically where in a particular database to begin retrieving or writing information then can be used in the query statement or in the query arguments to build a dynamic query, and they can be used in the End statement to tell the database what to do when the interaction is complete. Interpretations can also be used in the Prompt Line Meta-slot. Recall that the Prompt Line

Meta-slot specifies how to query the user for the value for a particular slot. Interpretations allow one to build a dynamic query for the user, giving him some information about the current state of the session.

B7 Pattern Matching

Another method of allowing one to test the values of slots without mentioning them explicitly is through the use of pattern matching. Pattern matching creates a list of objects which belong to a parent class or object. There are two basic types of pattern matching:

(I) The first type, called a universal qualifier, allows one to test conditions like: do all members of this class (or do all sub-objects of this object) meet this condition, and

(2) The second type, called an existential qualifier, allows the test: are there any members of this class (or any sub-objects of this object) which meet this condition.

A pattern matching always generates a list of objects to test or use, whether it is a pattern matching on a class or an object. Furthermore, this list is always the first level of objects reached on each branch of the object network. When it reaches an object on one branch it doesn't search down that branch further. When it reaches classes, it continues to search down the different branches until it reaches some objects. In existential pattern matching has a very important function besides being a test: it keeps the list of objects which meet the given criteria. This local list can be used in either subsequent LHS conditions or the RHS actions, but after the rule is completely evaluated then the list will be lost (of course if one wishes to save the list, the CreateObject operator can be used to attach it to another class, and subsequent pattern matching can operate on this class). If there is a rule which has more than one pattern matching, then subsequent pattern matching work on the result of the first pattern matching .A series of pure existential qualifiers is commutative, and a series of pure universal qualifiers is commutative, but a mix of the two is not commutative. Evaluating universal qualifiers first is more restrictive than using existential qualifiers first. Interpretations can be combined with pattern matching, providing even more flexibility. Interpretations are always nested within a pattern matching, never vice versa.

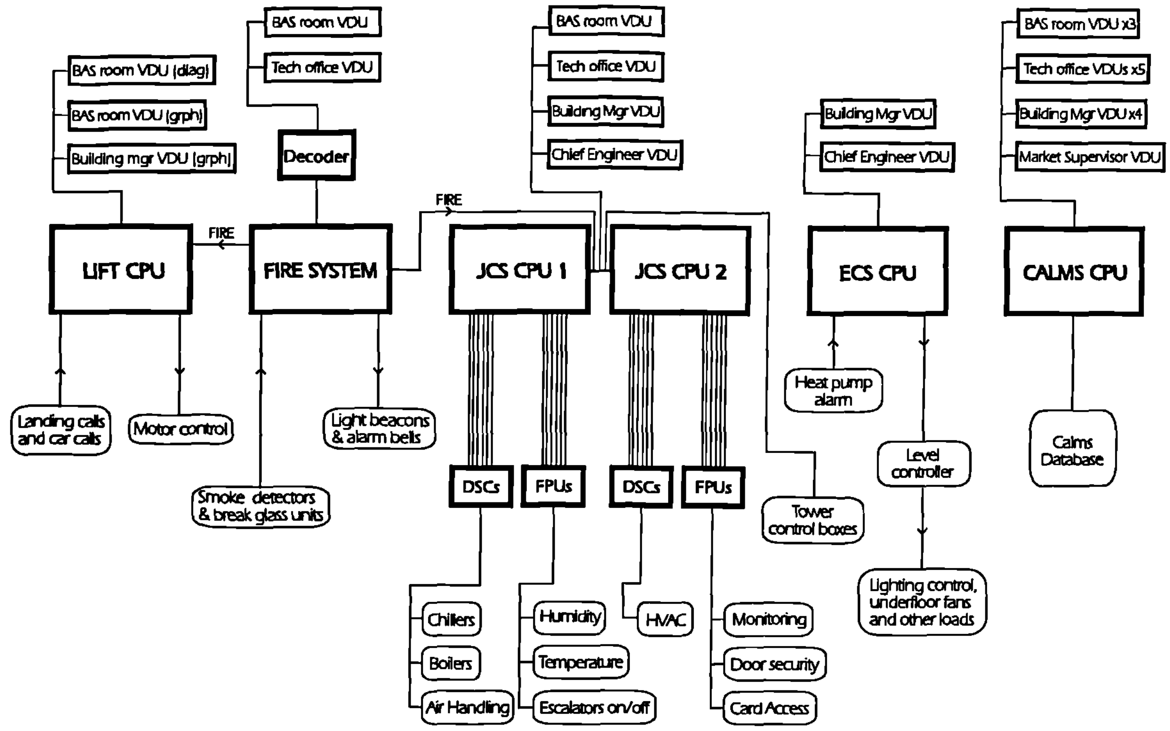
APPENDIX C - SOFTWARE SIMULATION

A software simulation was constructed of a IBMS central controller database. Data was obtained from the Lloyd's control and maintenance management systems and stored in the object-oriented structure as described in chapter three. This was used to investigate the control and facilities management rules processed on the central control system. A rule-based system was constructed to simulate the electrical load shedding and economy mode features given in chapter four. The rules include a wide range of plant equipment classes including lighting, pumps, computer equipment, chillers and air handling units.

This appendix gives an overviews of the Lloyd's systems around which the software simulation is based. The plant and building data-base is described in further detail along with the was in which Nexpert meta-slots were used to implement methods. The appendix concludes with an annotated Nexpert rule example taken from the facilities management knowledge-base.

C1 Description of the Lloyd's BMS

At the time it was opened in 1986 the Lloyd's building contained some of the most advanced building control systems in this country. Lloyd's has a direct digital controlled building management system. This consists of a computer controlled switching system and maintenance software package. An overview of the systems is shown in Fig.C1. The central monitoring for the BMS, lighting control and maintenance equipment is housed in the basement. The JCS 85/40 BMS computer system controls 45 air handling systems, 7MW of heating equipment, 4.2MW of cooling, 6MW of electrical input and 3MW of standby power generation. There are 48 intelligent and 63 non-intelligent outstations with the former capable of running independently in the event of a breakdown of the main BMS computer.



Express Lifts

12 wall climber lifts controlled by a standard algorithm. Includes fault diagnosis and status VDUs.

Hal Fire Systems

Main panel in the security control room. Smoke detectors, sounders, sprinklers and links with the JCS.

Johnson Control Systems

Building management system. Plant monitoring and control of HVAC direct digital control of plant performed by 35 digital system controllers (DSCs) and 63 field processing units (FPUs).

Energy Control System

The lighting and under floor equipment control system. Control of 8900 lights, 879 heat pumps and 1178 fans.

Contract Admin. System

Planned preventative maintenance and repair work of some 20,000 items of plant and machinery.

Fig.C1 Lloyd's Building Systems

The main boiler house contains two Robey Lintherm 3.5MW dual-fuel (oil and gas) fired boilers supplying mthw at 7 bar, and one 1.7MW boiler for summer use. The main oil tanks store 100 000 litres, which will keep the building fuelled for about ten days in the event of a gas shortage. Sequencing is automatically handled by the building management system. The main 11kV intake leads to eight units of 250MVA 630A vacuum breakers. All main distribution panels are then interconnected by bus-bars, and if power is lost they can link up the busbars and run half a board off an adjacent transformer. A 250kVA uninterruptible power supply has been installed to serve the computer equipment. Two rotary alternator sets, each running at 50% capacity, are driven from the mains electrical supply. A nickel cadmium battery bank, rated for 20 minutes duration, takes over in the event of mains failure. Emergency power generation is handled by two W H Allen 1750kVA diesel generator sets with full automatic start-up and load sharing capability. This is also supervised by the building's management system.

Chilled water is provided by three Hall water chillers with a combined cooling capacity of 4.2MW. The air conditioning takes the form of an under floor system and conditioned air is pushed out into the 300mm deep floor plenum from header ducts. Small fan units within the plenum take approximately a half mix of primary and recalculated air and deliver the mixed air via circular floor mounted grilles. A further energy control computer (ECS) controls the switching individually of all the 8500 main luminaires, 900 heat pumps and 1600 fan air terminals within the building. Each level of the building has its own master processor assembly which allows autonomous operation of each floor regardless of failures on other floors or the main ECS computer. The ECS computer is linked to the BMS as well as to the fire detection system so that fan air terminals can be shut off under alarm conditions.

There are 12 wall climber lifts, in three groups of four, and four fireman lifts controlled via a self-contained system utilising the manufactures standard algorithm. An animated display on a VDU shows the position and status of each lift together with car and landing calls. Up peaks are detected utilising weight detectors in the cars themselves. A data logger system identifies some 30 different breakdown codes and problems are immediately reported to the central control room.

C2 Plant Database

This section overviews the current control of Lloyd's services and explains how this control data is represented in the plant database.

The Lloyd's 1986 building has a complex lighting control system with 8,900 separate luminaires each under the control of the building's Energy Control System. The lights are grouped with each group of lights having an individual switching pattern switched by 285 autonomous lighting control boxes. Most of the lower floors contain around 760 lights but the number reduces on the higher floors as the building tapers. Each luminaire has a unique number, a location number, type, group and status as shown in table C1. The group number is the specific switching group for the luminaires within each area. For example, suppose there are 100 lights and 20 are assigned to switching group one, 60 to group two and 20 to group three; providing the members of each group are evenly spaced in an area, switching these groups provides 20%, 80% and 100% light level control.

Light	Location	Type	Block	Status
1	1201	1	1	off
2	1201	1	1	off
3	1201	1	1	off
..	1201	1	1	off
n	1909	3	2	on

Table C1 : Extract from the Lighting Data Base

The type field is used to cross reference against the *LUMTYPE* database. This stores luminaire specific data such as the power rating and maintenance data. The status field indicates whether or not a luminaire is 'on', 'off', or 'out of service'. These terms are defined later.

The electrical supply in each area is divided into three classes; essential loads, important loads and non important loads. This can be implemented by multiple ring mains on each floor. Sensitive main computer equipment and the telephone exchange can be plugged into the essential loads supply. VDUs and word processors are plugged into the important loads supply, leaving desk lamps and fan heaters plugged into the non important supply. Such a system is already implemented at Lloyd's but using only one additional ring main for computer equipment which is connected directly to the UPS systems.

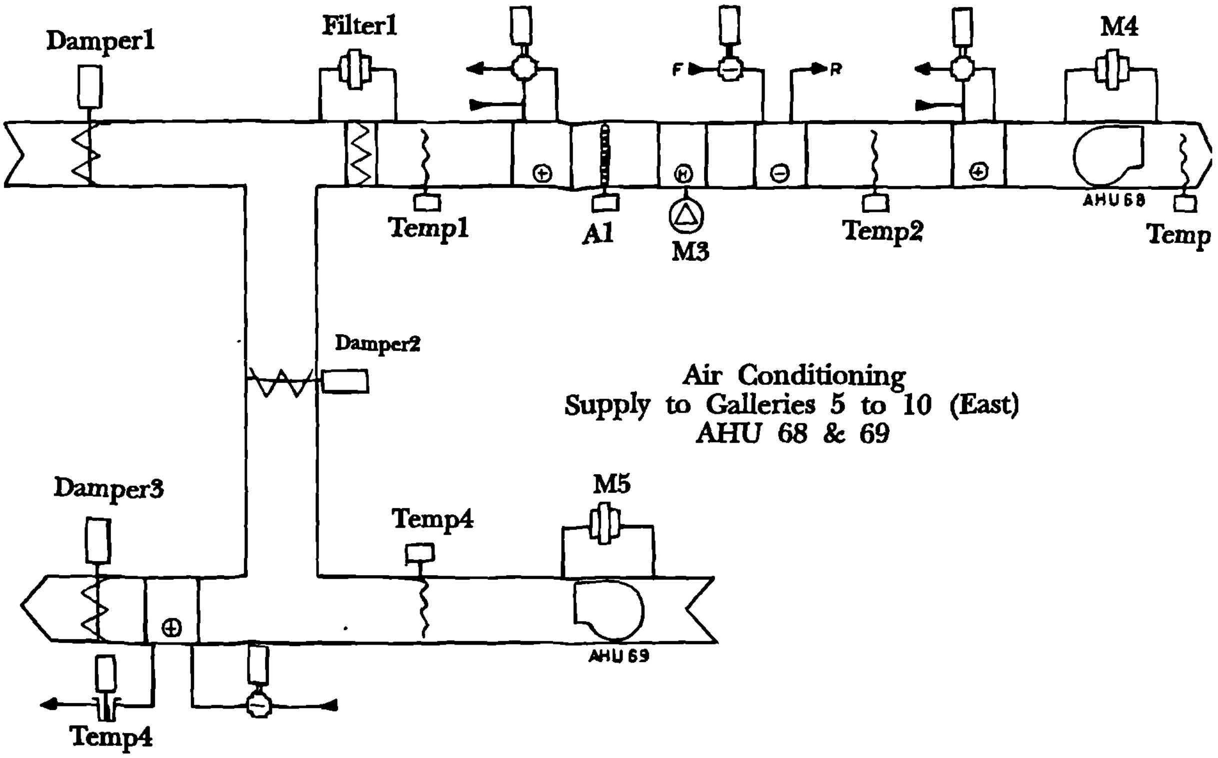


Fig.C2 Schematic of AHU69, 69 in the Lloyd's Building

Fig.C2 shows an air handling unit (AHU) from the Lloyd's building. This is typical of such systems in large buildings. There are two fans; one for supplying air into the areas (fan 68) and one for extracting (fan 69). The AHU incorporates a number of sensors which are connected to the Johnson's control system. T1 to T13 are temperature sensors and H1 is a relative humidity sensor used by the control system. There are also safety sensors such as a freeze protection thermostat, flow switch, and differential pressure switches over the filters. The AHU also has a humidifier in addition to the heating and cooling sections. This AHU is a mixed air control type handling unit, meaning that it uses both outside and return air. This increases the operational efficiency. Total return air would not be used as it would give very poor air quality. The air dampers are controlled such that the outside air damper and return air damper operate together, with the mixed air damper moving in exactly the opposite direction. This gives control over the air ratio and hence its quality. The system is controlled such that when the temperature of the outside air decreases the outside air flow is reduced to minimum before the heating valve is opened. A minimum set value, which depends on the area, must exist to ensure a minimal quantity of air. When an area is not occupied the outside air flow can be closed completely.

During the occupied cycle the room is maintained by controlling the temperature of the discharge air. This is done by the following sequence of actions: heating, ventilation and cooling. In the unoccupied times the fans are stopped, outside air dampers closed and the mixed air damper fully opened. If the temperature drops below a pre-set minimum the fans start and the heater valve is fully opened, thus preventing condensation in room. Some systems also have CO_2 monitoring for air quality control. In such schemes the fans are equipped with two speed controls and when high levels of CO_2 are measured, the fans run at higher speeds. Currently Lloyd's do not implement such a scheme[40].

The HVAC database, shown in table C2, differs from the other two in that the equipment, instead of being specific to a location, is mainly specific to an air handling unit which may feed many locations. The equipment itself may be quite remote from the area of the building it serves. For example, an air handling unit may be in the basement and form part of an air supply feeding the top floor of the building. Therefore, the database stores the AHU number along with the equipment's actual location number. HVAC equipment is further divided into many classes such as pumps, humidifiers, filters, boilers, Chillers, pumps and fans.

HVAC	class	no	function	AHU	location	type	status	power
27	AHU	3	generator room supply	3	LB air handling unit room 42A	1	off	0
28	AHU	19	UB general supply	19	LB air handling unit room 42A	2	off	0
29	AHU	20	UB general supply	20	LB air handling unit room 42A	3	off	0
30	AHU	24	boiler room supply	24	LB air handling unit room 42A	1	off	0
31	AHU	4	generator room exhaust	4	LB air handling unit room 42A	4	off	0
32	AHU	5	restaurant supply	5	LB air handling unit room 42B	2	on	200
33	AHU	6	restaurant exhaust	6	LB air handling unit room 42B	4	on	200
34	AHU	8	kitchen exhaust	8	LB air handling unit room 42B	3	on	200
35	AHU	25	LB corridor supply	25	LB air handling unit room 42B	5	on	200

Table C2 : Extract from the HVAC Database

On initialisation the Nexpert rule-base reads in the three databases and sets up an internal dynamic object structure. The sub-classes 'Elec', 'Lighting' and 'HVAC' inherit the properties *control_status* and *power_measured* from the parent class 'PLANT'. An object is created for each record in the electrical equipment database and linked to the 'Elec' sub-class. Each object inherits all the properties from the 'PLANT' class with the additional property *location_set* from the sub-class. Similar inheritance occurs in the case of lighting and heating objects.

The breakdown data is read into Nexpert on a 'need to know basis'. The Nexpert Retrieve operator is used to transfer records from an external database into working memory. Each record is retrieved into a set of slots belonging to a specific object. This operator supports a sub set of the structured query language to control which records are transferred. This enables selective transfer of data. For example, if the fault is diagnoses as a boiler fault only the past

---

[40] Lloyd's are looking at implementing such a scheme in the near future.

```
          INHERITANCE STRATEGY : Default
          INHERITANCE STRATEGY : Class first
          INHERITANCE STRATEGY : Breadth first
          ORDER OF SOURCES :
            Do         120              Input.time_period1
            Retrieve   "c:\occupan.slk"
          @TYPE=SYLKDB;@SLOTS=Input.occupancy1;@FIELDS="occupancy";@QUERY=
      "time_period = @V(input.time_period1)";@CURSOR=Input.retrieve;
            Do         140              Input.time_period2
            Reset      Input.retrieve
            Retrieve   "C:\occupan.slk"
          @TYPE=SYLKDB;@SLOTS=Input.occupancy2;@FIELDS="occupancy";@QUERY=
      "time_period = @V(input.time_period2)";@CURSOR=Input.retrieve;
            Reset      Input.retrieve
            Do         160              Input.time_period3
            Retrieve   "C:\occupan.slk"
          @TYPE=SYLKDB;@SLOTS=Input.occupancy3;@FIELDS="occupancy";@QUERY=
      "time_period = @V(input.time_period3)";@CURSOR=Input.retrieve;
            Do         Input.occupancy1+(Input.time_period-
      Input.time_period1)/20*(Input.occupancy2-Input.occupancy1)
          Input.predicted_occupancy
          IF CHANGE DO :
```

The @V operator in the query field is used to match the database time period field against the required time. The field occupancy is retrieved into the Nexpert global slot *input.occupancy1*. The *input.retrieve* slot is an integer property which is set to one if the retrieve is successful. Three such operations retrieve the value of occupancy at the first interval below the required time and the two above. These values are then used in the Gaussian function above to calculate the value of occupancy at the time period specified by the slot *input.time_period.*

C4 Knowledge Processing

The Nexpert agenda contains a prioritised list of hypotheses, not rules. In its quest to find the value of a hypothesis, one or many rules are often executed, but they are evaluated in order to find the state of the hypothesis. In order for Nexpert to reason, there must be at least one hypothesis on the agenda. One can explicitly place a hypothesis on the agenda by 'suggesting' it, or it can be put on the agenda as a relevant goal by one of the inference search mechanisms. The evaluation of a rule is the most basic event in knowledge processing. This consists of attempting to find the state of its hypothesis. When evaluating a rule, the default strategy is to evaluate the left hand side conditions from top to bottom. However, this default strategy can be modified by using the slot's inference priorities. The condition which has the slot with the highest inference priority is processed first, then the condition which has the second highest inference priority, and so on.

If a condition has several slots in it, then the highest inference priority of any slot is used for the conflict resolution. Inference priorities can also be dynamic. This means that an inference atom can be attached to any slot. When Nexpert evaluates the rule, the inference atom's current value becomes the slot's inference priority.

Several rules leading to the same hypothesis generate a logical 'or' graph using the logical states of the rules described above:

- All rules must be FALSE for the hypothesis to be FALSE
- At least one rule must be TRUE in order for the hypothesis to be TRUE
- At least one rule must be NOTKNOWN and no rules TRUE for the hypothesis to be NOTKNOWN

Nexpert reasons according to the closed-world assumption. This means that if the rules leading to a particular hypothesis are evaluated as FALSE, then Nexpert will conclude that hypothesis is FALSE as well and will use that determination in further reasoning. While the conditions within a particular rule are logically 'anded' together (i.e. all of them must be TRUE for the rule to be TRUE), multiple rules pointing to the same hypothesis are connected by 'ors'. Thus if any number of rules pointing to a particular hypothesis are evaluated as FALSE, and one is evaluated as NOTKNOWN then the hypothesis will be NOTKNOWN. If a rule pointing to a particular hypothesis is evaluated as TRUE, then the hypothesis will be TRUE.

When Nexpert evaluates a hypothesis with multiple rules leading to it, the rule with the highest rule priority will be evaluated first. Similar to inference and inheritance priorities, rule priorities can be either static or dynamic. If a rule has an Inference Priority Slot declared and its value is KNOWN, then the value of this integral slot will be used in conflict resolution to determine when this rule is evaluated. If a priority slot has not been declared, then the value of the priority number will be used. If several rules leading to the same hypothesis have the same inference priority, regardless of whether it is determined by a static or dynamic priority, then the rule which contains the condition with the highest inference priority on one of its slots will be evaluated first.

The energy management rule-base is divided into several rule-bases which are loaded into and out of memory as required. This saves working memory and speeds up inference time as only those rules which are relevant are in memory and hence processed. Initially the energy rule-base is loaded into Nexpert and initialised. This accesses the building, plant equipment and resources data bases and the class/object structure is set up in the form of dynamic objects. Rules are then fired to establish the activity of areas within the building based on the known and predicted occupancy. The next highest priority rules to fire are those which check to see if the building should be in a load shedding state. If this is the case the ranking rules are loaded from disk and processed to order the relative ranks of all the loads. When the hypotheses of all these rules are known the load shedding rules are loaded into memory. These are processed in the order of their inference priority slot thus ensuring than loads are shed based on their ranking order. When no further shedding rules can be fired due to the energy balance being achieved the second load shedding knowledge-base is loaded. This modifies the shedding criteria such that further loads are shed to make power available to engage high priority loads which, according to their ranking order should be on, but are currently off. After these are processed the load shedding rules are removed from memory and replaced with the load engage rules. These activate loads passing control to the zone energy management routines and PID controllers. These rules include load cycling control for peak demand reduction.

The Nexpert inference strategy makes use of the 'if change' meta slot. This enables actions to be performed if a change is made in the value of a slot. This change could be as the result of a calculation, a definition or from updated data read in from a file. At regular intervals the object data is updated from the resources and plant data bases. Any change in power demand, supply or building activity is detected by the appropriate meta slot which conducts a strategic reset of any rules which may be influenced by this change. For example, a change in the *input.power* slot evokes the reset of all load shedding rules (assuming they are in memory) and the hypothesis of the rule to check for a load shedding situation is loaded onto the agenda. Rules such as those used in the global control knowledge base to define times of the day are linked to a class called |GLOBAL| which is systematically reset every thirty minutes.

When there is a change in the data from a specific area such as an increase in occupancy the appropriate rule hypothesis are reset leaving all rules which are not affected by the change alone. This assures maximum rule processing efficiency and response time.

C5 NEXPERT Rule Example

The following is an annotated rule example from the facilities management problem diagnosis rule-base.

For the hypothesis problem to become true the left hand side conditions require the hypothesis *find_ahu* to be true. This causes backward chaining to rules which display a menu of floors for the user to select from. For example, in the case of the Lloyd's building system, if Gallery 6 is selected, all rooms on gallery 6 are listed and the operator selects the area by clicking on the appropriate name using the mouse.

```
If    find_ahu is TRUE
      And Input.ahuE_set is KNOWN
Then problem
      is confirmed.
      And |DATABASE|.directory is assigned to |DATABASE|.directory
      And load_Ahu is assigned to load_Ahu
      And "" is assigned to output
      And STRCAT(output,"the air supply comes into ") is assigned to output
      And STRCAT(output,Input.room) is assigned to output
      And STRCAT(output," from AHU ") is assigned to output
      And STRCAT(output,INT2STR(Input.ahuS_set)) is assigned to output
      And STRCAT(output," and the exhaust is AHU ") is assigned to output
      And STRCAT(output,INT2STR(Input.ahuE_set)) is assigned to output
      And STRCAT(output,". The supply comes from ") is assigned to output
      And STRCAT(output,'ahu'\Input.ahuS_set\.location) is assigned to output
      And STRCAT(output,".") is assigned to output
      And Execute "Message"(@STRING="@TEXT=@V(output),@OK";)
```

A meta-slot exists for this object which loads the data on air handling units from the HVAC database. The rule then builds up a text string in the object *output* to inform the operator which air handling units relate to the area in question and where they are located. There are two rules, one for where an area has a separate supply and extractor AHU and another where only a general supply is specific. The hypothesis is true if the sub goals *Find_ahu* is true and the *ahuE_set* property of the object input is known. This assumes that this rule is only evoked if the problem has already been located to a specific AHU thus forcing backward chaining via the *Find_ahu* hypothesis. The *ahuE_set* property stores the plant number of the extractor for this air handling unit. If this is not known then the area does not have a specific extractor as in the atrium of the Lloyd's building. In this eventuality it will look up the extraction fan from a related area stored in the 'Building' database. When the required sub system has been identified the building database is read in and a the class object structure is created.

C6 Ranking Rules

The following summarises the operation of some of the load ranking rules used in the emergency handling rule-base of the control knowledge-base.

Lighting 20% to 80%

If area is dormant & not important ⇒ rank becomes 975
If area is dormant & important ⇒ rank becomes 915
if area is quiet & not important ⇒ rank becomes 850

if area is quiet & important ⇒ rank becomes 810
if area is occupied or active & not important ⇒ rank becomes 810
if area is occupied or active & important ⇒ rank becomes 820

Lighting 0 to 20%

if area is dormant and non important rank ⇒ 950
if area is dormant and important rank ⇒ 800
if area is quiet and non important rank ⇒ 450
if area is quiet and important rank ⇒ 300
if area is (occupied or active) & non important rank ⇒ 400
if area is (occupied or active) & important rank ⇒ 250

Electrical non important

if area is (quiet or dormant) & important rank ⇒ 950
if area is (active or occupied) & rank ⇒ 750 (725)

Electrical important

if quiet,dormant rank ⇒ 850 (825)

if active,occupied rank ⇒ 650 (625)

Electrical Essential

if quiet,dormant rank = 250 (200)
if active,occupied rank = 200 (150)

HVAC AHU

if dormant rank = 1000 (900)
if quiet rank = 875 (825)
if active,occupied rank = 800 (700)

C7 Ventilation Requirements in Buildings

New ventilation guidelines have recently been published by the EEC. These a reported in a 1992 EEC publication entitled 'Guidelines for ventilation requirements in buildings, report no.11' and are summarised here.

The ventilation rate required for health and comfort should be calculated separately and the highest value used for design. The ventilation required for a health point of view is calculated by this equation:

$$Q_n = \frac{G}{C_i - C_o} . \frac{1}{\varepsilon_v}$$

where,

Q_n = ventilation rate required for health (l/s)
G = pollution load of chemical (μg/s)
C_i = allowable concentration of chemical (μg/l)
C_o = outdoor concentration of chemical at air intake (μg/l)
ε_v = ventilation effectiveness

The ventilation required for comfort is calculated by this equation:

$$Q_c = 10. \frac{G}{C_i - C_o} . \frac{1}{\varepsilon_v}$$

where,

Q_c = ventilation rate required for comfort (l/s)
G = sensory pollution load (olf) [Table C4,5]
C_i = perceived indoor air quality, desired (decipol) [Table C6]
C_o = perceived outdoor air quality at air intake (decipol)
ε_v = ventilation effectiveness

	Sensory pollution load olf (m^2floor)	
Existing buildings	**Mean**	**Range**
offices	0.3	0.02-0.095
schools	0.3	0.12-0.54
kindergartens	0.4	0.20-0.74
assembly halls	0.5	0.13-1.32

Table C4 Pollution load caused by the building, including furnishing

	Sensory pollution load olf/occupant
Stationary, 1 - 1.2 met[42]	
0% smokers[43]	1
20% smokers	2
40% smokers	3
100% smokers	6
Physical exercise	
low level, 3 met	4
medium level, 6 met	10
high level (athletes), 10 met	20

Table C5 Pollution load caused by occupants

Quality level (category)	Perceived air quality		Required ventilation rate l/s.olf
	% dissatisfied	decipol	
A	10	0.6	16
B	20	1.4	7
C	30	2.5	4

Table C6 Three levels of perceived indoor air quality (examples)

---

[42] One met is the metabolic rate of a resting person (1 met - $58W/m^2$ skin area, i.e. approx 100W for an average person).

[43] Average smoking rate 1.2 cigarettes/hour per smoker, emmision rate 44ml CO/cigarette

APPENDIX D - HARDWARE IMPLEMENTATION

A prototype zone controller incorporating an occupancy predictor was built using a stand alone 8086 processor card with digital and analogue input/output to process the data. This was linked to a 486 PC which operates as the central control computer running the global control rule-base. In addition to occupancy prediction, the controller uses an internal mathematical thermal model of the area to predict optimum start and stop times depending on environmental conditions.

D1 AS1-1 Analogue and Digital Power Interface

D1.1 Overview

The requirement is to control a number of machines and devices such as motors and to receive analogue inputs from light and temperature sensors. The PC is fitted with a hardware interface card called the AS-1 [92], and drives relays which controls the equipment. There are also two temperature sensors and two light sensors. The AS1 system is outlined below.

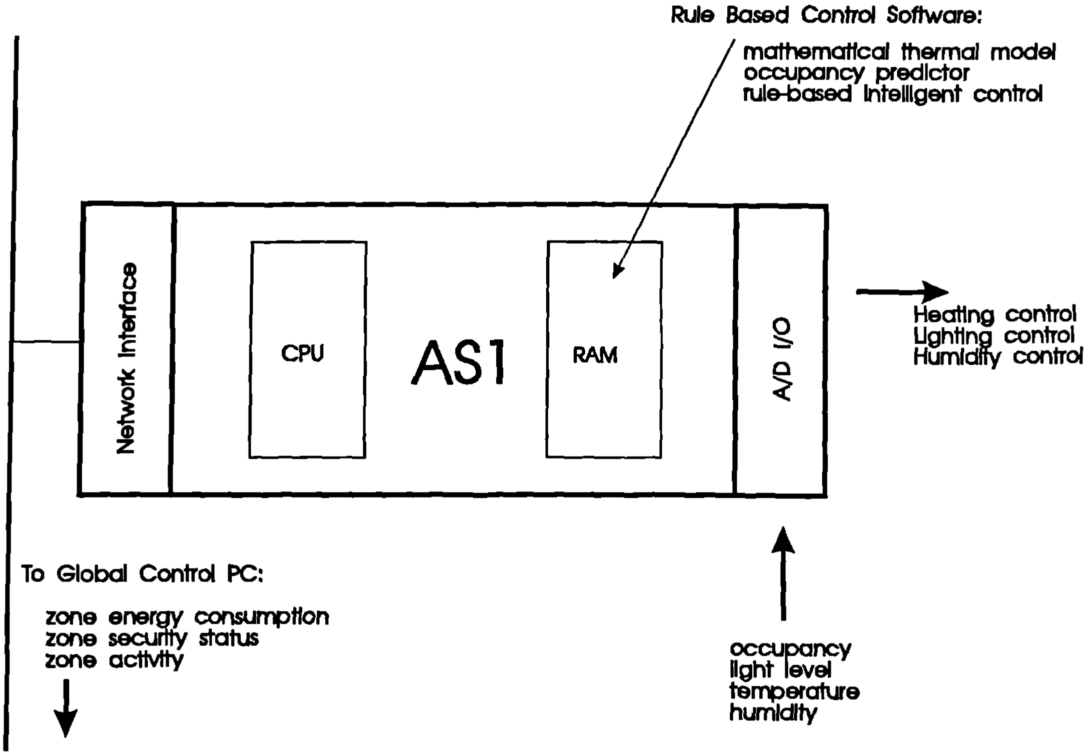


Fig. D1 AS-1 Prototype the System

The AS-1 comprises a complete computer system with its own memory and peripheral devices, linked via a flexible interface to another host computer system [Fig.D1]. On power-up or after a system reset, the AS-1 is in an idle state and application programs must be down loaded into its own memory. Once down loaded they run independently of the host computer but can communicate with the host system via shared buffer areas of memory. In this way it mimics the main IBMS and zone control sum-system relationship.

D1.2 Input/Output

The relay board consists of six relays each relay consists on two double throw contacts. They come complete with socket and safety cover. The switched output is wired on the assumption it will be switching mains voltage at 10A. The control is low voltage 30mA. Two of these are used for the heating pump 'open' and 'close' motor signals and the rest for lighting control. The sensors consist of the specific devices mounted on a small pieces of varo-board or similar (approx. 2in × 2in) with a 3 way plug/socket connection. The are connected to the analogue inputs using reference voltages from the AS-1.

Occupancy sensors, using two simple break beam type units, are fitted at the entrance to each area along with the light and temperature sensors. The system was implemented in room 43A at Brunel in collaboration with Brunel's estates department. This is a large machines lab and a classic example of a large area heated and lit daily although the scheduled occupancy was less than 8 hours per week.

The occupancy sensor direction detection was implemented in software using a simple algorithm to detect the sequence of beam breaks. A hardware implementation, shown in Fig.D2, was also devised. The circuit detects the direction of motion using two latches and simple logic. The outputs are passed to a digital counter and then to the zone controller. A hardware approach would prevent the zone controller being tied up in people counting.

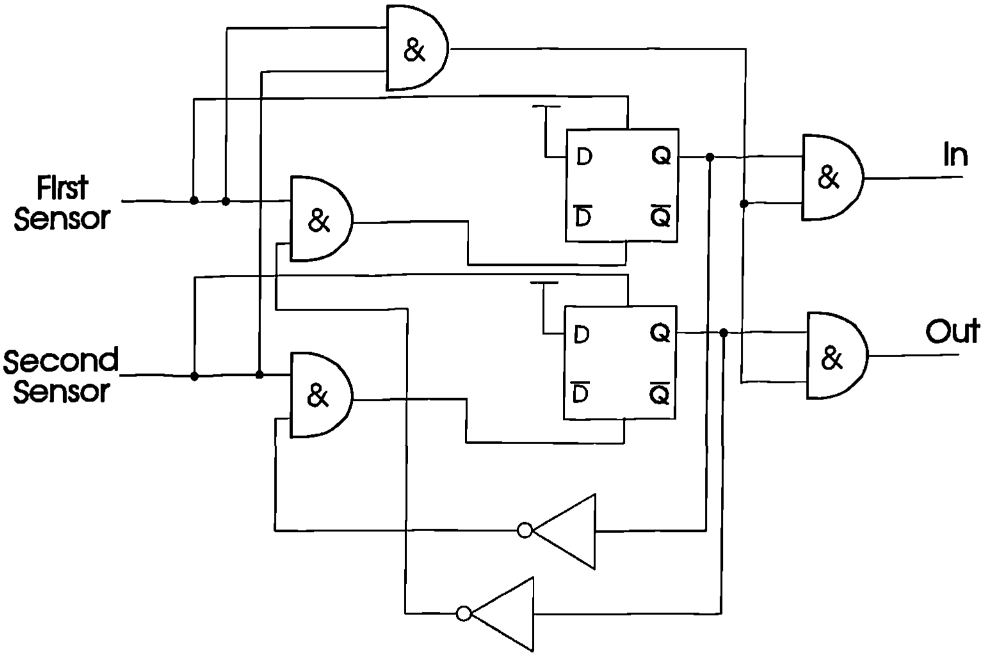
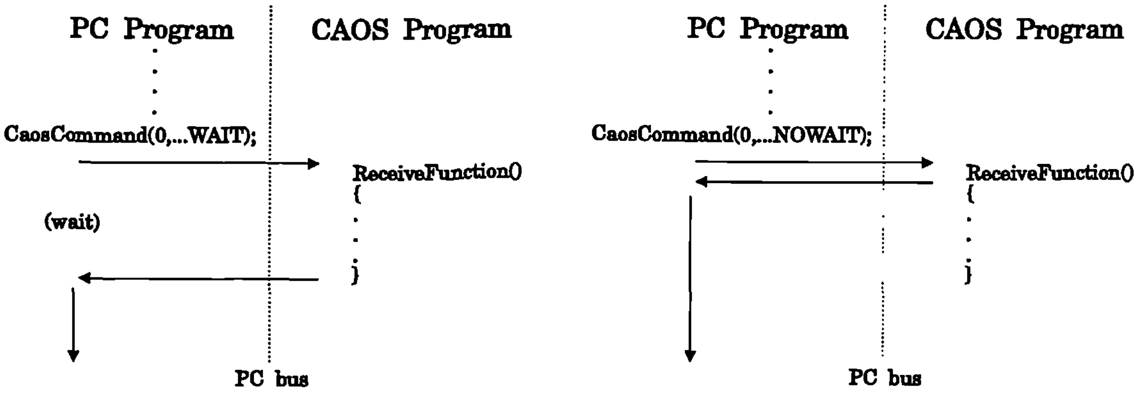


Fig.D2 : Digital Occupancy Sensor Direction Detector

D1.3 Communications

The computer can communicate with the controller board in two ways using a shared area of memory as shown in Fig.D3. The computer program can pass control to the CAOS board and suspend program execution until a response is received. Alternatively the computer program can pass a parameter to the CAOS board and the continue execution until a reply is received.



WAIT

NOWAIT

Fig.D3 : PC and CAOS Board Communication Timing Diagram

D1.4 Thermal Model Results

The thermal model was tested in a large room on the Brunel campus. The room is approximately 240,000 m^3 (40m× 50m ×120 m) in volume and is heated by water filled heating panels. Access was given to the Trend heating control system and the optimum start and stop times for this room were investigated during evenings and weekends in December 1992. The room is assumed to be lightweight in construction.

The room was allowed to cool and the measurements were made for the time to heat and cool between temperatures in the range 12 to 22 degrees. Measurements were made with the heating systems on at full power and for various external temperatures occurring during a period of two weeks. Measurements were taken within eight hours of heating (late on a weekday evening) and after 40 hours of cooling (on a Sunday evening). The first two results were used to calculate a value for the constants *a* and *b* [Table D1]. Pre-heat and cooling times calculated using these values were then compared with practical results.

t_1 = Time to heat from θ_{01} degrees to ϑ_1 degrees [s].
t_2 = Time to cool from θ_{02} degrees to ϑ_2 degrees [s].
ϑ_1 = Minimum room temperature at the start of occupancy [°C].
ϑ_2 = Minimum room temperature at the end of occupancy [°C].
θ_{01} = Initial room temperature at the optimum start time [°C].
θ_{02} = Room temperatures at the optimum stop time [°C].
θ_{e1} = External temperature during optimum start [°C].
θ_{e2} = External temperature during optimum stop [°C].

$$a = -t_2 \Big/ \ln\left(\frac{\vartheta_2 - \theta_{e2}}{\theta_{o2} - \theta_{e2}}\right) \quad \text{and} \quad b = \left(\frac{\theta_{o1}.e^{-t_1/a} - \vartheta_1}{e^{-t_1/a} - 1}\right) - \theta_{e1}$$

Calculating *a*,*b* from Experiment	θ_{01} Deg.C	ϑ_1 Deg.C	t_1 Sec	θ_{e1} Deg.C	θ_{02} Deg.C	ϑ_2 Deg.C	t_2 Sec	θ_{e2} Deg.C	*a* Sec	*b* Deg.C
Results 1	12	21	3300	2	21	15	1500	2	3952	25.9
Results 2	15	22	1980	7	22	17	1620	7	3995	25.9
1% change in *a*	12	21	3333	2	21	15	1515	2	3992	25.9
1% change in *b*	12	21	3218.2	2	21	15	1500	2	3952	26.2

Table D1 : Results used to Calculate *a* and *b*

The two results give an average value of 25.9 degrees for *a* and 3974 seconds for *b*. To see how critical the values of *a* and *b* are in predicting t_1 and t_2 the tables show calculated values for these times for a 1% increase in *a* and *b*. A 1% increase in *a* increases t_1 by 1% and t_2 by 1% which are both changes of less than 30 seconds. A 1% increase in *b* decreased t_1 by 2.5% and has no effect on t_2. Therefore the value of *b* is more critical, for the room in question increasing the predicted start time by nearly 5 minutes for a 1 degree change. Therefore, if values of *b* found from experiment are within 1 degree of each other one might assume an accuracy in the predicted start/stop time of less than 5 minutes.

Calculating t_1,t_2 from Experiment	θ_{01} Deg.C	ϑ_1 Deg.C	t_1 Sec (pred.)	θ_{e1} Deg.C	θ_{02} Deg.C	ϑ_2 Deg.C	t_2 Sec (pred.)	θ_{e2} Deg.C	t_1 Sec (act.)	t_2 Sec (act.)	error in t_1 %	error in t_2 %
Results 3	13	20	2304	3	21	15	1611	3	2400	1590	4.1	1.3
Results 4	11	22	3784	3	22	11	3437	3	3870	3510	2.3	2.1
Results 5	12	21	3314	2	21	15	1508	2	3450	1530	4.1	1.5
Results 6	15	22	3105	2	22	17	1143	2	3180	1170	2.4	2.3
Results 7	12	21	2567	5	21	15	1867	5	2610	1890	1.6	1.2
Results 8	15	22	2304	5	22	17	1384	5	2340	1410	1.5	1.8

Table D2 : Further Results used to Calculate *a* and *b*

Table D2 shows further results for various external temperatures, internal temperatures and cooling times. The predicted start/stop times are calculated using values of *a* and *b* calculated above. The maximum error between predicted and calculated is 4.1% (96 seconds) and the mean start error is 2.7%. For the building in question, the mathematical model predicted optimal start and stop times to within two minutes. The results show a good correlation despite the simplifications involved in the mathematical model. However, these results were obtained during a two week period, during which large variations in external temperature did not occur. Nevertheless, the self tuning aspect should keep the system in tune under varying conditions.

APPENDIX E - HEVASTAR SIMULATION

Predicted Energy Savings for Brunel's Central Lecture Block

The central lecture block (CLB) is a three story building consisting of 36 seminar rooms, 16 medium size lecture theatres and 6 large lecture theatres. During term time individual rooms are occupied for one hour periods from 8.45 in the morning to 21.00 at night.

E1 Predicted Heating Savings

Heating savings were predicted using a commercial package called HEVASTAR. This is a suite of software packages for building services engineers for computer aided design and design evaluation. A program was used to calculate the annual energy consumed in heating the building. This is done by first calculating the design heat loss of the building and then using a version of the degree day method, modified to enable permissible heat gains from solar, lights and people to be allowed for.

Two programs were used: 'HLOSS' which estimates the building's thermal energy loss and 'ENERGY' which calculates the annual heating energy consumption of the building. The building was classified into four classes of rooms. Table E1 shows the dimensions and lighting for these rooms.

Room Description	No of	Length (m)	Width (m)	Height (m)	lighting no of 120W tubes	Notes
1 - class room	36	10	10	2.9	5	1 wall 2/3 glass
2 - small lecture room	16	10	8.5	3.5	4 x 5 = 20	all internal walls
3 - large lecture outer	4	21	15	3.3	5 x 8 = 40	3 external walls
4 - large lecture inner	2	21	15	2.2	5 x 8 = 40	2 external walls

Table E1 : CLB Room Classification

The 'HLOSS' program was used to calculate the heat loss for individual rooms for a forced air heating system. All classrooms are heated by radiators and have cavity wall insulation in their external walls. All lecture theatres have forced air ventilation, the class rooms are naturally ventilated. The lecture theatre ventilation rate is approximately 4 air changes per hour and the ventilation system does not incorporate heat recovery. Table E2 shows the U value constants used[44].

Component	U value ($Wm^{-2}K^{-1}$)
105mm solid brick wall	3.3
220m solid brick wall	2.3
335mm solid brick wall	1.7
260mm cavity brickwork	1.2
150mm solid concrete	3.4
2mm airspace double glazing	2.9
6mm single glazing	5.6

Table E2 : Typical U Values

The following analysis gives the heat losses for individual rooms:

ROOM 1 Room (36 off)

```
      Surface                          Area      U    Temperature   Fabric Loss

                                       (m2)    value   difference    (Watts)


  Exp sed WALL type 1                  29.00    1.20      22.0          761

  Int. PARTITION type 1                29.00    2.30       0.0            0

  Int. PARTITION type 2                29.00    2.30       0.0            0

  Int. PARTITION type 3                29.00    2.30       0.0            0

  Internal CEILING                    100.00    2.30       0.0            0

  Internal FLOOR                      100.00    2.30       0.0            0

                                                                     -------

                                                       FABRIC LOSS      761

  Volume    290.00 m3  Air change rate  1.00  Natural INFILTRATION     2165

                           Height Allowance for  2.90m  ( 2.9%)          85

                                                                     -------

                                                  TOTAL HEAT LOSS      3011 W

                                                                     -------



                           - Temperatures -

                 Resultant                 22.0

                 Environmental             21.9
```

---

[44] Callaghan P., Energy Management, McGraw-Hill, 1992

```
                    Air                      22.4

                    Mean Radiant             21.6


                            - System -

               Forced warm air downward from high level


                    ------ ----- --------------------

                    TOTAL ROOM HEAT LOSS    3011 W

                    -- -----------------------------


                         Average    10.38 W/m3

                                    30.11 W/m2
```

ROOM 2 (16 off)

```
     Surface                            Area      U    Temperature   Fabric Loss

                                        (m2)    value   difference     (Watts)


  Int. PARTITION type 1                29.00     2.30      0.0             0

  Int. PARTITION type 2                23.20     2.30      0.0             0

  Int. PARTITION type 3                29.00     2.30      0.0             0

  Int. PARTITION type 4                23.20     2.30      0.0             0

  Internal CEILING                     80.00     2.30      0.0             0

  Internal FLOOR                       80.00     2.30      0.0             0

                                                                       -------

                                                        FABRIC LOSS        0

  Volume    297.50 m3  Air change rate  0.00  Natural INFILTRATION         0

                       Air change rate  4.00  Forced   VENTILATION      8727

                           Height Allowance for  3.50m  ( 3.5%)          305

                                                                       -------

                                                     TOTAL HEAT LOSS    9032 W

                                                                       -------


                          - Temperatures -

                    Resultant                22.0

                    Environmental            22.0

                    Air                      22.0

                    Mean Radiant             22.0
```

```
                          - System -

                Forced warm air downward from high level


                   ------  --------------------------

                   TOTAL ROOM HEAT LOSS      9032 W

                   ----------------  -----------------


                        Average   30.36 W/m3

                                 106.26 W/m2
```

ROOM 3 (4 off)

```
      Surface                         Area      U    Temperature   Fabric Loss

                                      (m2)    value   difference     (Watts)


  Exposed WALL type 1                69.30     1.20      22.0          1771

  Exposed WALL type 2                49.50     1.20      22.0          1265

  Int. PARTITION type 1              69.30     2.30       0.0             0

  Int. PARTITION type 2              49.50     2.30       0.0             0

  Exposed ROOF                      315.00     1.20      22.0          8048

  Internal FLOOR                    315.00     2.30       0.0             0

                                                                     -------

                                                       FABRIC LOSS    11083

  Volume    1149.75 m3  Air change rate  0.00  Natural INFILTRATION       0

                        Air change rate  4.00  Forced   VENTILATION   36990

                               Height Allowance for  3.65m  ( 3.7%)    1755

                                                                     -------

                                                   TOTAL HEAT LOSS    49827 W

                                                                     -------


                          - Temperatures -

                    Resultant                  22.0

                    Environmental              21.3

                    Air                        24.1

                    Mean Radiant               19.9


                             - System -
```

```
                        Forced warm air downward from high level


                           -  ---------  --------------------

                              TOTAL ROOM HEAT LOSS    49827 W

                                 ---  --  --------------------


                                 Average   43.34 W/m3

                                          158.18 W/m2
```

ROOM 4 (2 off)

```
    Surface                              Area      U    Temperature   Fabric Loss

                                         (m2)    value   difference     (Watts)


 Int. PARTITION type 1                  69.30    2.30       0.0            0

 Exposed WALL type 1                    49.50    1.20      22.0         1271

 Int. PARTITION type 2                  69.30    2.30       0.0            0

 Int. PARTITION type 3                  49.50    2.30       0.0            0

 Internal CEILING                      315.00    2.30       0.0            0

 Solid GROUND FLOOR                    315.00    1.20      22.0         8089

                                                                      -------

                                                       FABRIC LOSS      9361

 Volume    1149.75 m3  Air change rate  0.00  Natural INFILTRATION         0

                       Air change rate  4.00  Forced   VENTILATION     36483

                            Height Allowance for  3.65m  ( 3.7%)       1673

                                                                      -------

                                                    TOTAL HEAT LOSS    47517 W

                                                                      -------


                                - Temperatures -

                           Resultant               22.0

                           Environmental           21.4

                           Air                     23.8

                           Mean Radiant            20.2


                                  - System -

                       Forced warm air downward from high level
```

```
---------------------------------

 TOTAL ROOM HEAT LOSS    47517 W

---------------------------------


          Average   41.33 W/m3

                   150.85 W/m2
```

This gives a total heat loss for the CLB of 547kW.

The Hevastar 'ENERGY' program was then used to find the effect on the building's energy consumption of changing the occupancy time.

Economic benefits of intermittent heating based on,

- Thermal response of the plant
- The thermal response of the building
- The duration of heating, of cooling and pre-heating
- The relative capital and running costs.

This means reduced running costs are the result of an increase (and hence costs) .(i.e. large boilers and heat emitters and hence high capital costs)

The solar heat gains (Qs) are computed for each window in the building, using data from the CIBSE energy code. The light and people heat gain Q_{LP} is computed:

$$Q_{LP} = Q_L + Q_P$$

The maximum value permitted for Q_{LP} is $10W/m^2$. If the system control does not incorporate room thermostats, then Q_{LP} is set to zero. For a system with one thermostat per room/zone and not shaded, the total heat gain Q_t is set to $Q_S + Q_{LP}$. For a system with no thermostats, Q_S is set to zero. For all other thermostat arrangements, the solar gains QS are computed as is they were north facing glazing and then the total heat gain Q_T is set to Q_S+Q_{LP}.

The degree day value and the length of the heating season (L) in weeks are used to compute the average outside temperature (T_O) during the heating season.

$$T_o - 15.5 - D_D/L/7$$

The f value (F_R) for the building is found from:

$$F_R = (A_Y + V)/(A_U + V)$$

This is then used with the heating control system information to compute the average preheat period (P_H) for the building. The preheat period is based on a plant ratio of 1.2, it is then corrected for the actual plant ratio specified. The average temperature difference for the building (D_T) can then be computed:

$$D_T = (H + P_H)\ RF\ (T_I - T_O)/(H\ R_F + (24-H))$$

The heat gains can then be deducted from the heat losses, and the occupancy days/weeks (W) used to compute the total annual heating energy requirements (E) for the building.

$$E = L\ W\ 24 \times 3600\ (Q\ D_T/(TI - T_W) - Q_T)$$

Calculations of energy consumption are carried out monthly for a typical day in each month, using average monthly conditions. Fresh air amount can be fixed or variable. If fixed, then the % re-circulation is specified. If variable, the minimum percentage fresh air is specified, if known otherwise the program will compute this from the fresh air requirements. Duct gains are taken into account. Pump electrical energy is computed using a pump transport factor. HWS energy consumption can optionally be included.

The results are shown in table E3 and illustrated in Fig.E1

Hours Occupation	**Predicted annual energy consumption (GJ)**
7	2743
8	2863
9	2973
10	3074

Table E3 : Predicted Annual Heating Energy Consumption for CLB

In general, results show that the annual saving can be made of around 3.7% from 1 hours reduction in heating and 7.7% for 2 hours from a normal 9 hour occupancy period. Energy savings depend of the type of room, its occupancy patterns and the time of year. Obviously higher savings can be made in sporadically occupied rooms where HVAC set points can be relaxed for more than an hour and when the heating at start up is at full capacity to reduce the optimum start time.

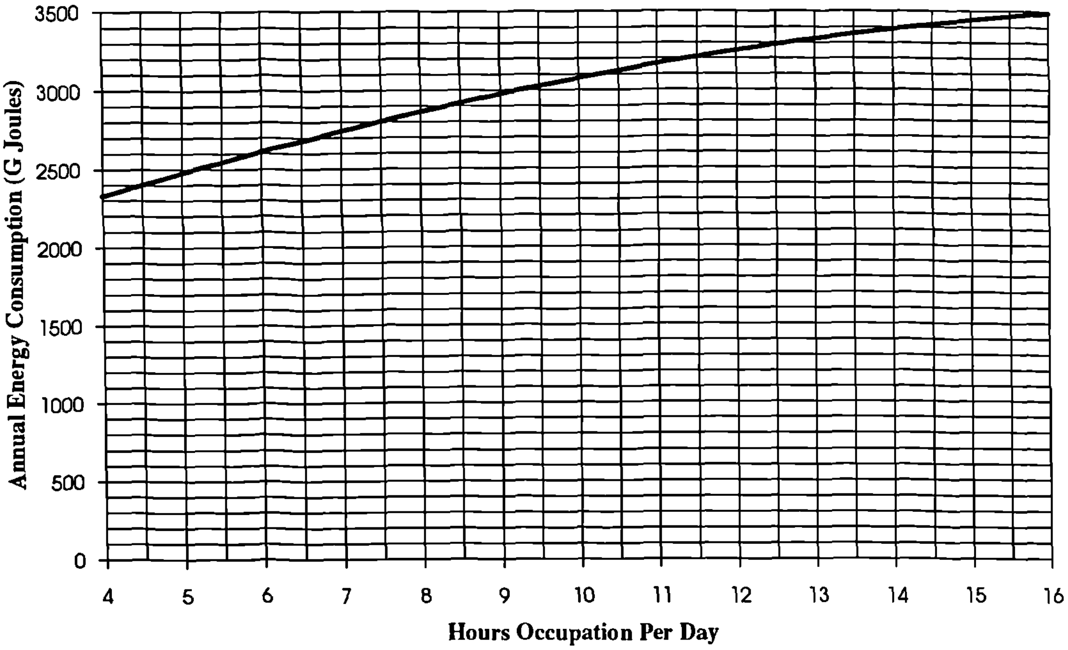


Fig.E1 : Predicted Annual Energy Consumption

Lectures are currently scheduled between the hours of 9.00 and 6.00. One can assume that for a typical lecture theatre, the actual occupancy will be less than the full 9 hours. The rooms are not occupied for one hour at lunch time and there is often at least one unoccupied period in the morning or afternoon. A reduction in occupancy time of two hours a day could give rise to an annual heating energy saving of 7.7% [table E3].

```
Type of fuel : Gas
             : Cost 1.341p/kWh (39.3p/Therm)
Plant + System seasonal efficiency  90 %


Building annual energy consumed   3302.63  Giga Joules
Annual fuel cost                 12302.71  pounds
```

(This compaies well with the annual estimated fuel budget for the CLB of £12,000 - estates department, Brunel)

```
Predicted savings = 7.7% of £12302.00
        = £947.00
```

E2 Electricity savings

Occupancy sensing could save electricity in terms of lighting and ventilation. During peak times, Brunel buys electricity at the cost of 4.12p per kWh + V.A.T. An increased tariff is charged for the two months around Christmas.

E2.1 Lighting

Each of the class rooms has 5 lights, the small lecture theatres have 20 and the large lecture theatres each have 40 fluorescent tubes. If each tube is 120W, the building's total lighting load is 88kW. This comprises 180 lights for class rooms, 320 lights for the small lecture theatres and 240 tubes for the large lecture theatres.

If one assumes that during a normal day half of the lighting is in use between normal lecture hours which are 8.45 to 5pm. No lectures take place between 13.00 and 14.00 and each room has at least one other free period per day, therefore lighting could be reduced by 2 hours per room during this time. The CLB normally remains open until 21.00 and has some use during the evenings. If one assumes that a fifth of the lighting remains on unnecessarily during this time it is possible that this could be reduced to as little as 10% during the hours 17.00 to 21.00. An approximate saving in lighting electricity can be calculated assuming 251 working days in the year (365 - 104 days for weekends and 10 public holidays).

Current predicted lighting electricity consumption:

88000 x 0.5 x 251 x 8 = 88352kWh
88000 x 1/5 x 4 x 251 = 17670kWh
TOTAL = 106022kWh
Annual cost = £5132

With occupancy sensing:

88000 x 0.5 x 251 x 6 = 66264kWh
88000 x 1/10 x 4 x 251 = 8835kWh
TOTAL = 75099
Annual cost = £3635

£1496.00 predicted savings per annum (29% saving in lighting electricity)

E2.2 Predicted Ventilation Electricity Saving

Room	Supply fan	Extract fan	Ventilation
4 x 4 small theatres	8.7A 3Φ	8.5A 3ϕ	$5m^3/s$
Theatres A,B & C	4.3A 3Φ	4.3A 3ϕ	$3m^3/s$
Theatres D & F	4.3A 3Φ	4.3A 3ϕ	$3m^3/s$
Theatre F	6.2A 3Φ	4.3A 3ϕ	$4m^3/s$

Table E4 : Ventilation Data

Table E4 shows the power rating for the supply and extractor fans which ventilate the CLB building.

total ventilation electricity = 52.3kW
2 hour a day saving using load control
32300 x 2 x 251 =16214kWh saving

Total predicted cost saving = £785.00

E3 Summary

Heating savings = £947.00
Lighting Savings = £1496.00
Ventilation Savings = £785.00

TOTAL predicted annual savings = £3228.00

If we assume that a simple occupancy sensor could be designed and manufactured for £100 per unit and that it takes 1 hour to wire it into the existing Trend system at £30/hour labour charge, the cost per room would be £130. The CLB has 58 lecture rooms which gives a total installation cost of £7540.

This could mean that such a system, integrated with the existing TREND controllers, could pay for itself in:

7540 / 3228 = 2.3 years or 28 months.

APPENDIX F - BOILER FAULT DIAGNOSIS RULES

F1 Fault Diagnosis Example : Boiler Fault Diagnosis

F1.1 Alarm Situations

```
If   Input.class_set is "boiler"                          RULE 6187
     And symptom10a is not FALSE
     And symptom17 is TRUE
     And boiler_alarm is "excess temperature alarm?"
     And fault10 is FALSE
     And fault13 is FALSE
Then fault9 is confirmed.
     And fault9.fault is assigned to Input.fault
```

Rule 6187 diagnoses a temperature problem with the boiler. The 'sympton10a is not FAULT' clause is used to include the possibility NOTKNOWN as the value of symptom 10a. There are many faults that may not be deemed to be a fault. For example, in this case the boiler will probably be running so an inexperienced operator may answer 'no' to the question 'is there a fault?'. The rule makes use of rules 6102 and 6110 to check the status of faults 10 and 13. This causes backward chaining to these rules to ensure their hypothesis are false. If the excess temperature alarm has gone off and the fault is not due to excess pressure or the flow valve does not need adjusting the problem could be due to an excess temperature alarm fault. This may mean a failure of the thermostat or sensing circuitry. For fault9 to be confirmed the problem must be due to the circuitry associated with the boiler and thus is a job for the contract engineers (PME). Further rules to diagnose problems with the boilers are outlined below.

```
If   Input.class_set is "boiler"                          RULE 6109
     And symptom10a is not FALSE
     And symptom17 is TRUE
     And boiler_alarm is "Oil select status point alarm?"
     And symptom11 is not TRUE
Then fault12 is confirmed.
     And fault12.fault is assigned to Input.fault
```

Rule 6109 checks to see if an oil status point alarm has occurred using symptom17. If it has and it is the oil select status point alarm and it is not a control problem the advice is to check the oil level. Symptom 17 is used to differentiate this as a fault that requires an alarm to be confirmed.

```
If   Input.class_set is "boiler"                          RULE 6167
     And symptom10a is not FALSE
```

```
        And symptom13a is TRUE
        And symptom17 is TRUE
        And boiler_alarm is "smoke density alarm?"
        And symptom8a known
        And symptom8 is TRUE
        And symptom18 is not TRUE
    Then fault7
        is confirmed.
        And fault7.fault is assigned to Input.fault
```

If the smoke density alarm has gone off, and the mixture has definitely been adjusted for clean combustion and the smoke density meter has not cleaned recently rule 6167 gives advice to clean the smoke density meter. The density of the smoke is usually measured by light sensor measuring the light level form a source through the smoke stream. A common problem is for the sensor to get sooted up and read too low a light level thus triggering the smoke density alarm. The symptom8a is known test is included to load symptom 8a on the agenda to prompt likely related rules should fault7 be found false. Symptom8a is not tested as there may or may not be dark smoke. The fact that the alarm has gone off does not necessarily mean that it has been detected if the alarm is not functioning correctly.

```
    If  Input.class_set is "boiler"                      RULE 6119
        And symptom10a is not FALSE
        And symptom13a is TRUE
        And symptom17 is TRUE
        And boiler_alarm is "smoke density alarm?"
        And symptom8a known
        And symptom8 is TRUE
        And symptom18 is TRUE
        And symptom8a FALSE
    Then fault21
        is confirmed.
        And fault21.fault is assigned to Input.fault
```

Rule 6119 follows on from rule 6167. If there is a smoke density alarm and the boiler has been readjusted for clean combustion and the meter has been cleaned, assuming the smoke is not dark, there is no real problem with the boiler but with the alarm itself.

```
    If  Input.class_set is "boiler"
        And Symptom10a TRUE
        And Symptom13a is not TRUE
        And symptom17 is TRUE
        And boiler_alarm is "boiler lockout alarm?"
        and symptom11 is not true
        and symptom16 is false
        And symptom8b is false
        And symptom9 is false
        And symptom23 is false
    Then fault4 is confirmed.
        And fault4.fault is assigned to Input.fault
```

There exists a lockout alarm problem if the boiler is not running, the lockout alarm has occurred and it is not due to the boiler controls, Johnson controls, pressure correct and not the lockout relay failure.

F1.2 Symptom Problem Diagnosis

The following rules are not due to alarms but based purely on symptoms.

```
If   Input.class_set is "boiler"                    RULE 6107
     And symptom10a is not FALSE
     And symptom13a is TRUE
     And symptom9 is not FALSE
     And symptom12 is not TRUE
Then fault10 is confirmed.
     And fault10.fault is assigned to Input.fault
```

Rule 6107 diagnosis a pressure problem with the boiler. The rule assumes that there may be a fault and the boiler is running but the pressure is incorrect. It checks to see if the flow valve needs adjusting. If not, fault is due to incorrect boiler pressure and needs further investigation by the appropriate engineer.

```
If   Input.class_set is "boiler"                    RULE 6110
     And symptom10a is not FALSE
     And symptom13a is TRUE
     And symptom12 is not FALSE
Then fault13
     is confirmed.
     And fault13.fault is assigned to Input.fault
```

Rule 6110 reads 'this could be a fault, the boiler is running and the pressure is incorrect'. It checks to see if the flow valve needs adjusting, if so the advice is to adjust the flow valve for correct pressure/temperature. This again evokes the if change meta-slot and the appropriate class of engineer and adjustment time is read from the fault code database.

```
If   Input.class_set is "boiler"                    RULE 6108
     And symptom10a is TRUE
     And fault3 is FALSE
     And fault8 is FALSE
     And symptom23 is not FALSE
Then fault11 is confirmed.
     And fault11.fault is assigned to Input.fault
```

Rule 6108 checks to see if boiler is not running and if the problem is not with the boiler controls or the Johnson's controls then the problem may be due to the lockout relay.

```
If   Input.class_set is "boiler"                RULE 6111
     And symptom10a true
     And symptom17 is FALSE
     And symptom13 is not FALSE
     And symptom 11 is FALSE
Then fault14 is confirmed.
     And fault14.fault is assigned to Input.fault
```

Rule 6111 is true if there is a fault but no alarm has occurred and there may be a problem with the shut down point but not a control problem. The action is to investigate the problem with or reset the shut down point.

```
If   Input.class_set is "boiler"                RULE 6129
     And symptom10a is TRUE
     And symptom11 is not false
     And fault14 is FALSE
     And symptom23 is not true
     And fault8 is FALSE
     And symptom8b is not TRUE
Then fault3 is confirmed.
     And fault3.fault is assigned to Input.fault
```

Rule 6129 reads 'if a fault has occurred, and there may be a problem with controls (there are the boiler's own controls and the JCS systems] and it is not a shut down point, not lockout relay failure, not JCS, not clean combustion (or may be), therefore boiler controls are at fault'.

```
If   Input.class_set is "boiler"                RULE 6151
     And symptom10a is TRUE
     And symptom13a is FALSE
     And symptom17 is FALSE
     And symptom16 is not FALSE
     And fault8 is FALSE
     And fault3 is FALSE
Then fault5 is confirmed.
     And fault5.fault is assigned to Input.fault
```

Rule 6151 reads 'the boiler is not running but there are no alarms, it is not running in auto and it is not a control problem therefore there is a problem with the auto mode setting'.

```
If   Input.class_set is "boiler"                RULE 6177
     and symptom10a is not false
     And symptom16a TRUE
     And symptom11 is TRUE
     And symptom16 is TRUE
     And symptom8b is not FALSE
Then fault8
     is confirmed.
```

```
        And fault8.fault is assigned to Input.fault
```

Rule 6177 reads 'a fault is suspected, the boiler is running in manual, not running in auto, there could be a control problem, it could be JCS (first this time), and not controls then it is a boiler control problem.

```
If   Input.class_set is "boiler"                RULE 6116
     And symptom10a is not false
     And symptom13a is TRUE
     And symptom17 is FALSE
     And symptom8a is TRUE
     And symptom18 is not TRUE
     And symptom8 is FALSE
Then fault1a is confirmed.
     And fault1.fault is assigned to Input.fault
```

Rule 6116 reads 'if there may be a fault and the boiler is running, no alarm has occurred but the smoke seems too dark, an adjustment to the controls of improved combustion is recommended'. Symptom 18 is not true test is just to check that the smoke density meter does not need cleaning, i.e. it has not already been cleaned.

```
If   Input.class_set is "boiler"                RULE 6101
     And symptom10a is not false
     And symptom13a is TRUE
     And symptom8a is TRUE
     And symptom8 is not true
Then fault1 is confirmed.
     And fault1.fault is assigned to Input.fault
```

Rule 6101 is not necessarily a fault, could just be an observation. This rule asks the uses if the boiler is running. If it is running and the smoke is dark and the has not recently been adjusted it suggests action 1 which instructs the users to adjust the mixture for clean combustion. This differs from the other scenario as it does not have to disprove fault7 as fault 7 will only occur if the alarm occurs and the mixture has not been adjusted.

```
If   Input.class_set is "boiler"
     And symptom17 is FALSE
     And symptom 21 TRUE
Then fault19
     is confirmed.
     And fault19.fault is assigned to Input.fault
```

This is a no alarm situation and the gas booster needs resetting.

F1.3 Non Fault Situations

As outlined above, the action does not have to be in response to a fault. The operator needs to know what class of engineer to send to change boiler settings or control settings. The following are included to show how non fault type actions integrates with the fault diagnosis rules for the boiler class rule-base.

```
If    Input.class_set is "boiler"
      and symptom 10a is FALSE
      And symptom14a is TRUE
      And symptom14 is TRUE
Then fault15 is confirmed.
      And fault15.fault is assigned to Input.fault
```

In this rule a fault is not suspected an thus symptom10 is false. Symptom14 is tested to see if a seasonal adjustment is required. In this case a change to summer mode operation is required. This requires adjustments of the boiler controls by a PME engineer.

```
If    Input.class_set is "boiler"                RULE 6113
      And symptom10a is FALSE
      And symptom14a is FALSE
      And symptom20 is TRUE
Then fault18 is confirmed.
      And fault18.fault is assigned to Input.fault
```

Rule 6113 is similar to the rule above but in this case it is not a seasonal adjustment but a change from gas to oil fuel. Lloyd's is a large consumer of gas and occasionally at peak times the gas supplier requests Lloyd's to temporarily change to oil supply. Lloyd's boiler's are dual fuel and a plateful supply of oil is stored in the lower basement. Oil will also be used in the event of a gas supply failure.

```
If    Input.class_set is "boiler"                RULE 6112
      And symptom10a FALSE
      And symptom14a is FALSE
      And symptom19 is TRUE
Then fault17 is confirmed.
      And fault17.fault is assigned to Input.fault
```

Safety regulations state that the condition of the boiler is inspected at regular intervals. This involves removing the boiler doors and inspecting the boiler rods for cracks and other signs of wear. This is checked by rule 6112.

```
If    Input.class_set is "boiler"                RULE 6118
      And symptom10a is FALSE
```

```
    and symptom14a is false
    And symptom13a is TRUE
    And symptom22 is TRUE
Then fault20 is confirmed.
    And fault20.fault is assigned to Input.fault
```

Rule 6118 may be fired if there is no fault on the boiler itself, no seasonal adjustment but the boiler needs to be shut down for maintenance on other heating equipment.

```
If  Input.class_set is "boiler"                RULE 6117
    And symptom10a is not FALSE
    And symptom17 is not FALSE
    And boiler_alarm is "Not known"
    And Symptom11 is not TRUE
Then fault2 is confirmed.
    And fault2.fault is assigned to Input.fault
```

Rule 6117 is included in the event that an alarm has occurred but the type is not known. It instructs the user to find the alarm code before proceeding. The test in symptoms10a and 17 and therefore include not known.

F2 List of Symptoms for all Plant Classes

SYMPTOM	DESCRIPTION
symptom1	"has the luminaire failed?"
symptom10	"Has the lockout relay failed?"
symptom11	"Is there a problem with the controls?"
symptom12	"Does the flow valve need adjusting?"
symptom13	"Is there a problem with the shut down point?"
symptom13a	"Is the boiler running?"
symptom14	"Does the boiler need to be put into 'Summer mode operation'"
symptom15	"Do boilers need to be put back on gas?"
symptom16	"Are the boilers running in auto?"
symptom17	"Has an alarm occurred?"
symptom18	"Does the boiler smoke density meter need cleaning?"
symptom19	"Do the boiler rod tubes need to be removed for inspection?"
symptom2	"has the starter failed?"
symptom20	"Do the boilers need to be changed from gas to oil?"
symptom200	"Is the area too warm?"
symptom201	"Is the area too stuffy, lacking air movement?"
symptom202	"Is the humidity in the area incorrect?"
symptom21	"Does the gas booster need resetting?"
symptom22	"Do the boiler(s need to be shut down for maintenance on some other equipment?"
symptom23	"Has there been a lockout relay failure"
symptom23a	"Is the chiller running?"
symptom23b	"Has a major part such as the motor just been replaced?"
symptom24	"Do the boilers need to be warmed through ready to put them into winter mode?"
symptom25	"Is there a pump problem?"
symptom26	"Is there a machine fault on the chiller?"
symptom27	"Check/adjust the chiller operation?"
symptom28	"Is there a problem with the CAM timer?"
symptom29	"Is this an initial check following a motor replacement?"
symptom3	"has the electricity supply failed?"
symptom30	"Do the JCS cables need to be disconnected for major part replacement?"
symptom31	"Do the JCS cables need to be reconnected following a major part replacement?"
symptom32	"Is there a leaking valve?"
symptom33	"Has there been a l.p. cut out?"
symptom35	"reinstate tower heating circuit?"
symptom36	"Is there a leak?"
symptom37	"Is the control valve gland faulty?"
symptom39	"Has the fuse blown?"
symptom39a	"Is the area temperature incorrect?"
symptom39b	"Is the fan running?"
symptom39c	"Is there a problem with a valve on the AHU?"
symptom4	"has the circuit breaker failed?"
symptom40	"Is there a broken valve on this air handling unit?"
symptom41	"Is the problem due to the AHU supply fan?"
symptom42	"Could there be a problem with the DEW point control?"
symptom43	"Does the equipment run in auto mode?"
symptom44	"Is the equipment locked off due to maintenance already in progress?"
symptom46	"Is there a fault with the cooling V/V?"

SYMPTOM	DESCRIPTION
symptom47	"Does the valve have a faulty gearbox?"
symptom50	"Is there a fault with the humidifier?"
symptom51	"Do the valves need to be opened to preheat coils on AHU?"
symptom52	"Does the flexible conduit to the thermostat need to be reinstated?"
symptom53	"Does a large part such as a motor need to be changed on the AHU?"
symptom54	"Is there a water leak?"
symptom59	"Is there water in the tank?"
symptom6	"has a second luminaire failed nearby?"
symptom60	"Do empty nitrogen bottles need to be replaced in the boiler room?"
symptom61	"Is the gauge glass broken?"
symptom62	"Do the run lights need replacing?"
symptom63	"Does the boiler system need resetting?"
symptom64	"Does the nitrogen need topping up?"
symptom65	"Is there a blockage?"
symptom66	"Has a fuse blown?"
symptom67	"Check the pump for correct operation?"
symptom68	"Is the timer defective?"
symptom69	"Does the motor need to be disassembled?"
symptom7	"has more than two luminaires failed nearby?"
symptom70	"Does the DSC alarm need to be reset?"
symptom72	"Does a new motor need fitting?"
symptom73	"Does a new pump need fitting?"
symptom74	"Does a new contractor need fitting?"
symptom76	"Is the pump stop locked?"
symptom77	"Do the indicator lights need replacing?"
symptom78	"Does the o ring need replacing?"
symptom79	"Does the pump need returning to auto mode?"
symptom8	"Has the boiler mixture been adjusted for clean combustion?"
symptom80	"Is the seal leaking?"
symptom81	"Does the pump require a service"
symptom82	"Shut valve to vent plant heating system?"
symptom83	"Is there leaking oil on the floor?"
symptom83a	"Has there been an oil leak?"
symptom83b	"Is the problem due to the motor rather than the pump it?"
symptom84	"Is there a starter fault?"
symptom86	"Is the temperature control of the heat pump low?"
symptom87	"Do the reservoirs need topping up?"
symptom88	"Is the pump tripping in alarm?"
symptom89	"Does the hydro constant need turning up?"
symptom8a	"Is the smoke from the boiler too dark?"
symptom8b	"Is the problem due to the Johnson's Controls?"
symptom9	"Is the pressure incorrect?"
symptom90	"Is there a valve fault?"
symptom91	"Is there a fault with the condenser water small by-pass valve?"
symptom92	"Check operation on the immersion heaters?"
symptom93	"Is there a loose belt?"
symptom94	"Do contractors require assistance when working on cooling towers?"
symptom95	"Do chlorine tablets and bromine need to be added to each pond?"
symptom96	"Has there been a DSC memory failure?"

F3 List of Fault Objects for all Plant Classes

class	code	description	trade	%	hour
boiler	1	adjust mixture to give clean combustion	pme	1.54%	4.00
boiler	2	boiler alarm	pme	15.38%	4.24
boiler	3	boiler controls	pme	4.62%	10.50
boiler	4	boiler lockout alarm	pme	36.92%	0.96
boiler	5	boilers not running in auto	pme	6.15%	1.50
boiler	6	check operation	pme	1.54%	0.58
boiler	7	clean boiler smoke density meter	pme	1.54%	2.00
boiler	8	control problem	jcs	1.54%	2.00
boiler	9	excess temperature alarm	pme	3.08%	0.88
boiler	10	incorrect pressure	pme	1.54%	1.00
boiler	11	lock out relay failure	jcs	1.54%	3.00
boiler	12	oil select status point alarm	pme	1.54%	2.50
boiler	13	open\close flow valve	pme	1.54%	1.00
boiler	14	problems with shut down point	pme	1.54%	4.00
boiler	15	put boiler into service in summer mode	pme	1.54%	4.50
boiler	16	put boilers back on gas	pme	1.54%	2.00
boiler	17	remove boiler rod tubes for inspection	pme	1.54%	17.50
boiler	18	request to change boilers from gas to oil	pme	3.08%	4.50
boiler	19	reset gas booster	pme	1.54%	0.41
boiler	20	shut down boilers for other maintenance	pme	3.08%	0.79
boiler	21	smoke density alarm	pme	6.15%	0.63
boiler	22	warm through ready to put into winter mode	pme	1.54%	2.75
chiller	23	pump down problem	pme	1.15%	5.00
chiller	24	machine fault on chiller	pme	82.76%	0.65
chiller	25	check/adjust chiller loading operation	jcs	2.30%	2.25
chiller	26	chilled water alarm	pme	1.15%	1.00
chiller	27	cam timer	pme	1.15%	3.50
chiller	28	water flow rate fault alarm	pme	4.60%	2.00
chiller	29	check initial starting after motor replacement	pme	1.15%	1.00
chiller	30	disconnect jcs cables	jcs	1.15%	2.50
chiller	31	reconnect jcs cables	jcs	1.15%	2.50
chiller	32	repair leaking valve	pme	2.30%	10.00
chiller	33	l.p. cut out	pme	1.15%	0.50
HE	34	high temperature alarm	jcs	20.00%	6.00
HE	35	reinstate tower heating circuit	pme	20.00%	0.50
HE	36	leak	pme	40.00%	5.83
HE	37	faulty control valve gland	jcs	20.00%	5.00
AHU	38	blocked filter alarm	pme	36.62%	1.59

class	code	description	trade	%	hour
AHU	39	blown fuse	pme	4.23%	1.92
AHU	40	broken valve on AHU	pme	1.41%	1.50
AHU	41	check AHU GAC supply fan	pme	4.23%	1.67
AHU	42	check DEW point control	pme	1.41%	1.00
AHU	43	check running in auto	jcs	1.41%	2.00
AHU	44	equipment locked off due to maintenance already	pme	1.41%	0.50
AHU	45	fan tripped alarm	pme	25.35%	0.75
AHU	46	fault with cooling V/V	pme	1.41%	1.50
AHU	47	faulty gearbox of valve	jcs	1.41%	4.50
AHU	48	fresh air damper alarm	jcs	1.41%	2.00
AHU	49	frost alarm	pme	1.41%	0.50
AHU	50	humidifier fault	pme	2.82%	1.58
AHU	51	open valves to preheat coil on AHUs	pme	1.41%	0.75
AHU	52	reinstate flexible conduit to thermostat	jcs	1.41%	0.50
AHU	53	remove motor from AHU	pme	1.41%	1.75
AHU	54	water leak	pme	9.86%	2.87
PU	56	alarm	pme	21.43%	0.41
PU	57	high pressure alarm	pme	14.29%	2.21
PU	58	low pressure alarm	pme	14.29%	2.58
PU	59	no water in tank	pme	7.14%	12.00
PU	60	replace empty nitrogen bottle in boiler room	pme	7.14%	1.00
PU	61	replace gauge glass	pme	7.14%	2.00
PU	62	replace run lights	pme	7.14%	0.66
PU	63	reset boiler system	pme	7.14%	1.00
PU	64	top up nitrogen	pme	14.29%	0.96
P	65	blockage	pme	1.49%	1.50
P	66	blown fuse	pme	1.49%	1.50
P	67	check operation	pme	7.46%	0.78
P	68	defective timer	jcs	2.99%	0.50
P	69	disassemble motor	pme	1.49%	4.00
P	70	reset dsc alarm	jcs	0.01%	0.00
P	72	fit new motor	pme	1.49%	2.00
P	73	fit pump	pme	1.49%	13.00
P	74	fitted new contractor	pme	1.49%	3.00
P	75	low pressure	pme	1.49%	0.75
P	76	pump stop locked	pme	4.48%	0.50
P	77	replace indicator light	pme	1.49%	0.50
P	78	replace 'o' ring	pme	1.49%	4.25
P	79	return to auto operation	pme	1.49%	0.50
P	80	seal leaking	pme	8.96%	1.88
P	81	service	pme	8.96%	1.79
P	82	shut valve to vent plant heating system	pme	1.49%	3.00

class	code	description	trade	%	hour
P	83	spread oil granules over leaking oil	pme	1.49%	1.00
P	84	starter fault	jcs	1.49%	1.00
P	85	switch pumps back to auto mode	jcs	4.48%	0.44
P	86	temp control of heat pump low	jcs	1.49%	2.00
P	87	top up oil reservoirs	pme	17.91%	1.10
P	88	tripping in alarm	pme	20.90%	1.07
P	89	turn hydro constant up	pme	1.49%	0.50
P	90	valve fault	pme	1.49%	8.00
C	90	fault with condenser water small by-pass valve	jcs	12.50%	6.00
C	91	check operation of immersion heaters	pme	12.50%	0.50
C	92	belts loose	pme	25.00%	0.75
C	93	assist contractors working on cooling towers	pme	12.50%	4.00
C	94	add chlorine tablets and bromine to each pond	pme	25.00%	1.38
C	95	DSC memory failure	jcs	12.50%	0.66

APPENDIX G - ARTIFICIAL NEURAL NETWORKS

G1 Overview of Neural Networks

Neural nets are a new and exciting area of artificial intelligence. They can replace traditional artificial intelligence techniques in many applications and their built-in 'fuzzy logic' ability enables them to solve hitherto unconsidered problems. They also offer a dramatic improvement in development time over conventional artificial intelligence. This makes neural net solutions commercially feasible where other artificial intelligence methods have not been. Neural networks offer possibilities for solving problems that require pattern recognition, pattern mapping, dealing with noisy data, pattern completion, associative lookups, and systems that learn or adapt during use. In addition, neural networks can perform some knowledge processing tasks. Some optimisation tasks can be addressed with neural networks and the range of potential applications is impressive. Neural nets are dramatically different from other software tools in their use. They require no programming in the conventional sense and dispense with most of the analysis of the problem which forms such a large part of conventional programming. They do, however, have their own requirements and their own set of considerations for successful implementation.

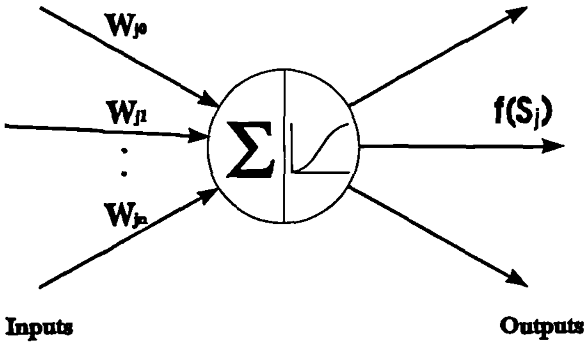


Fig.G1 A Neuron Processing Unit

Fig.G1 depicts an example of a typical processing unit for an artificial neural network. On the left are the multiple inputs to the processing unit, each arriving from another neuron, which is connected to the unit shown at the centre. Each interconnection has an associated connection strength, given as w,, w,, . . ., w,,. The processing unit performs a weighted sum on the inputs and uses a non-linear threshold function, f, to compute its output. The calculated result is sent along the output connections to the target cells. The most frequently used functions are the threshold and Sigmoid functions given by,

$$f(net_i) = \begin{cases} 1 & (true) & if\ net_i > 0 \\ 0.5 & (unknown) & if\ net_i = 0 \\ 0 & (false) & if\ net_i < 0 \end{cases}$$

$$f(net_i) = \frac{1}{1 + \exp(-net_i)}$$

Where, $net_i = \sum_{i=0}^{n} W_{ij} O_i$

The weights are stored in the node structure itself in a cyclic order. Various algorithms exist, using different strategies, for a node to calculate a new value. Neurons are subdivided into three types: Input, Output and Hidden. Hidden neurons are merely neurons that are internal to the neural net. Input neurons accept data from outside the neural net but perform no processing. These units assume the values of a pattern, represented as a vector. Hidden and output neurons perform almost identical processing, but only output neurons transmit data to outside the neural net. The middle, 'hidden' layer, of this network consists of 'feature detectors' which are units that respond to particular features that may appear in the input pattern. Sometimes there is more than one hidden layer. The network shown in Fig.G2 has three layers of neurons which is the typical organisation for the neural net paradigm known as back-error propagation.

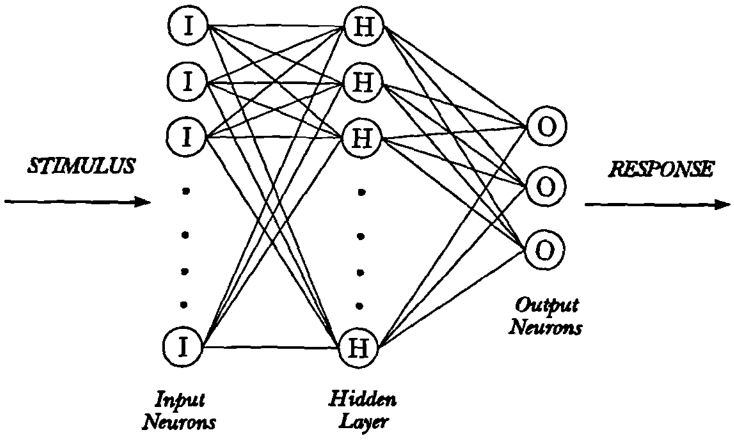


Fig.G2 The Artificial neural network

The neural net structure used in this work is the feed forward network with training using the back propagation learning rule. For each presentation of an input and output pair, the back propagation algorithm performs a forward calculation step followed by a backward propagation of the error. In the forward step the input vector is fed into the input layer and the net computes an output vector using the existing set of interconnection weights. The back propagation step begins with the comparison of the network's output pattern to the target vector. An error value for each output neuron is calculated based on this difference, and the weights are adjusted for all the interconnections to the output layer. Error values for the hidden layer preceding the output layer are then calculated. The weights of the hidden layer are updated, and the process is repeated until the first hidden layer is reached. Rumelhart and McClelland [93] give a good description of the back propagation algorithm.

The process of training neural nets is an iterative one. The training patterns are presented over and over again to the neural net and as each pattern is presented the neural net is asked for its response. The difference between this response and the target pattern is computed and the weights of the connections are modified to produce a better result next time. Incrementally, the average error for the whole set of patterns (the set of patterns is known as an epoch) is reduced. This process is called convergence. Normally, training is terminated by setting a limit for this average error. The neural net is said to be trained when it gives answers within this limit for the whole epoch. Occasionally, but less frequently with larger neural nets, the neural net will fail to converge no matter how long one waits. The problem can be resolved by redefining the input/output associations required or by using different training algorithms. Alternatively, one can attempt to split the problem into a number of smaller independent sections.

G2 NuralDesk

NuralDesk is a neural network software suite that provides an integrated development facility within the Microsoft Windows environment. It comprises tools for modelling, training and running neural networks, suitable for analysts, developers, researchers and other users wishing to integrate neural net capability into their applications. The components of the system are:

- NeuDesk - the problem interface program that enables one to design, train and run a neural network automatically from a set of data input in spreadsheet format.
- NeuModel - a network editor with which one can manually design, create and edit neural networks.

- NeuRun - the run time processor, controlled by external programs such as a database or spreadsheet, which supervises the processing and interfacing of the neural net.
- A number of dynamic link libraries combining the neural net algorithms, the code that performs the mathematical operations on neural nets.

The networks investigated in this thesis used Excel to store the training and interrogation data and passed this to NeuRun using Windows dynamic data exchange. For further information the reader is referred to the NuralDesk user's guide.

G3 Fault Code Prediction using Neural Networks

A number of network configurations were considered and training was conducted using different training algorithms for various numbers of hidden neurons. The neural network simulator used for the prototype uses floating point numbers for all its operations and the inputs and the outputs must be of this format. Furthermore, the inputs and outputs must always be in the range 0.0 to 1.0. A single input neuron is used to represent the time. This is included as some faults may be related to seasonal events. For example, heating related faults are more common in the winter as the equipment will get more use at this time of the year.

The network shown in Fig.G3 was constructed to investigate the correlation between the breakdown data, the time of year and the past breakdown history for different air handling units. If a fault code has occurred in the previous four weeks a '1' is applied to the stimulus input. For example, if the date is mid winter and a fan has tripped out, assuming there has been blocked filter alarms in the previous 4 weeks on this air handling unit, all this data can be used a training stimuli.

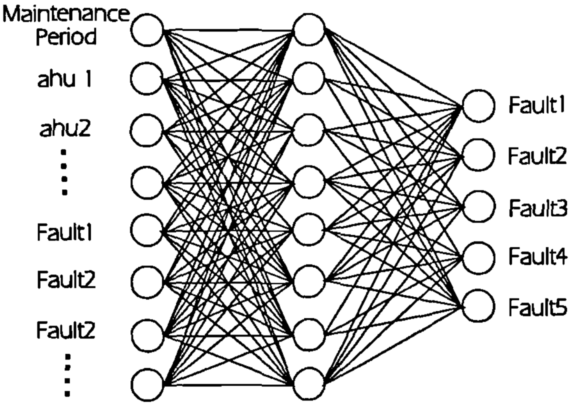


Fig.G3 Neural Network Fault Prediction

This network was trained to investigate correlation between previous fault history and fault prediction. The limitation here is that there is no way to represent more than one of the same faults occurring in the previous period. Furthermore, the training data does not take into account how recently the previous breakdown was. The network in Fig.G4 attempts to overcome this problem.

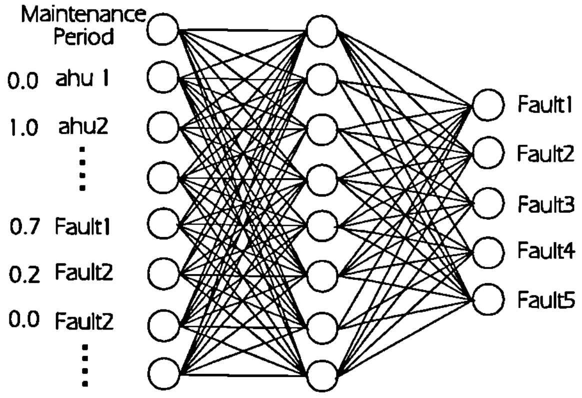


Fig.G4 Fault Prediction with Temporal Reasoning

If a past breakdown occurred three weeks ago, the fault input stimulus is set to the reciprocal of the whole number of weeks. The significance of the fault input value thus decreases with time. If the same fault occurs again in the same period this input is set to one again. A further variation would be to use a function on the input which scales the value to between one and zero to take account of faults which occurred more than four weeks ago.

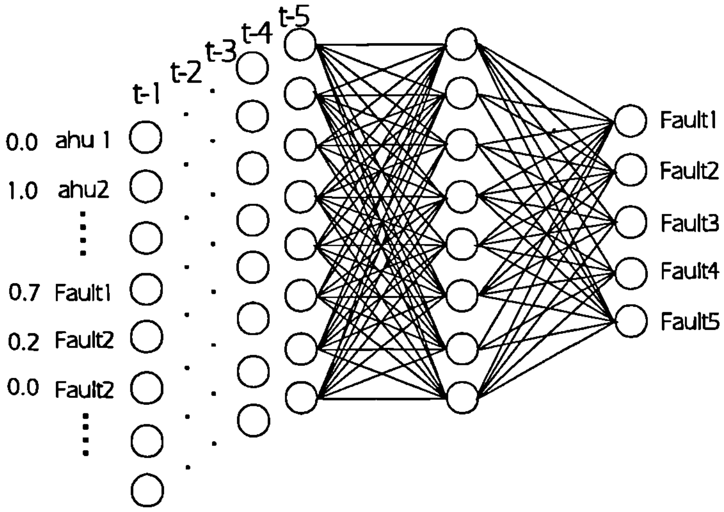


Fig.G5 Neural Network with 'Moving Window' of Inputs

The final approach investigated a moving window of all past faults for the last 3 weeks. This is illustrated in Fig.G5. Here a time period of one week is represented by a shift in the inputs and the problem of multiple faults of the same type in the previous weeks is thus solved. A value of 0.75 on the input indicates that the fault has occurred once this week and 1 indicates more than once. All the networks were trained using a year's breakdown data from the Lloyd's CALMS system. Most of the data was used for training but some was reserved for interrogating the network.

Initially the number of neurons in the hidden layer was set to equal the number natural logarithm of the number of inputs. For example, for 36 inputs 5 hidden neurons were used. This was recommended in the literature supplied with the NeuralDesk software. Variations in this were also tried. Feed forward from the neurons in the input layer to the outputs was also tried for one and two hidden layers. The networks were trained using three different algorithms, the standard back propagation (SBP), stochastic back propagation (BP) and quick propagation (QP). Researchers have been attempting to decrease the time taken to learn and to increase this ability of neural nets to generalise. The world is still waiting for the perfect algorithm, but it has been found that to improve generalisation you should not over train.

The networks were trained for 300 epochs as it was found that the error settled down to a minimum value after this many iterations. The results of training are given in tables G1 to G3 and in each case the average error is given. A very low training error was not expected as there are inconsistencies in the data. As can be seen in the tables, in all cases that the back propagation algorithm produced the lowest training results. The general conclusion is that the lowest training error is obtained by keeping the network simple. Little if any improvement is gained by a large number of hidden neurons or more than one hidden layer. The more layers of hidden neurons the network has increases the training time. It can be seen that extending the network to two hidden layers significantly increases the training error. However, the feeding forward of data from the input to the output layer reduced the training error in each case. The lowest training error was obtained with a single layer network with minimal neurons and feed forward. About one fifth of the available fault data was kept for interrogating the network to test its prediction ability using this network. In each case the error turned out to be slightly below the training error.

For networks 1,3 and 4 the training error was 7.6%, 8.2% and 9% respectively. The moving window of training stimulus in network 3 does give a lower training error but the interrogation response error is higher. This is probably because it is training against too much input data. Neural nets have the ability to interpolate and make an educated guess based on past experience but if over trained they loose this ability. In general one can conclude that there is some degree of correlation between the fault history and the neural network can be used to make a 'fuzzy' prediction for plant breakdown probability.

Network configuration 2 was chosen using the back propagation algorithm as this previously gave the lowest prediction errors. The air handling unit diagnosis rules use 18 symptoms and 4 alarm signals. This required adding 22 further inputs to the network and the number of hidden neurons was therefore increased to 8. The network was trained to predict the fault code, the repair time (as a fraction of 3 hours) and the class of maintenance engineer (1 = PME, 0 = Lloyd's). The results are given in chapter five.

Fault 1 (300 epochs)	BP (0.9,0.1)	SBP	QP
1 layer 5 neurons	0.135975	0,177452	0.211693
2 layers 5 neurons	0.147623	0.220307	0.224117
1 layer more neurons	0.118400	0.176874	0.207659
1 layer 5 neurons feed forward	0.123213	0.144683	0.213912
1 layer more neurons, feed forward	0.121997	0.144767	0.193531
2 layers 5 neurons feed forward	0.131684	0.144456	0.194562

Table G1 : Training and Stimulus Results for Network Fault 1

Fault 3 (300 epochs)	BP (0.9,0.1)	SBP	QP
1 layer 6 neurons	0.117883	0.193035	0.209453
2 layers 6 neurons	0.147712	0.220307	0.223805
1 layer more neurons	0.104924	0.173082	0.207083
1 layer 6 neurons feed forward	0.101788	0.124821	0.193051
1 layer more neurons feed forward	0.099426	0.124728	0.192307
2 layers 6 neurons feed forward	0.118434	0.220307	0.223805

Table G2 : Training and Stimulus Results for Network Fault 3

Fault 4 (300 epochs)	BP (0.9,0.1)	SBP	QP
1 layer 8 neurons	0.052416	0.076576	0.085823
2 layers 8 neurons	0.089104	0.088819	0.106025
1 layer more neurons	0.061692	0.090986	0.090270
1 layer 8 neurons feed forward	0.039186	0.081380	0.050002
1 layer more neurons feed forward	0.061906	0.051515	0.081250
2 layers 8 neurons feed forward	0.051761	0.088303	0.0534700

Table G3 : Training and Stimulus Results for Network Fault 4

APPENDIX H - LIGHTING MAINTENANCE CASE STUDY

H1 Case Study - Lighting

The Lloyd's 1986 building has a complex lighting control system with 8,900 separate luminaires each under the control of the building's Energy Control System (ECS). The lights are grouped by ECS with each group of lights having an individual switching pattern switched by 285 autonomous lighting control boxes. Each light in the building is digitally controlled, that is to say it is either turned on or off. There is no means of fading the light output from an individual luminaires. However, this can be done by switching only a fraction of all the lights on a specific area - i.e. every other light on would give a 50% light level. For example, some floors have a gradual increase in the light level at the start of the day by switching 50% of the lights on at 7am and the rest at 8am. A gradual increase in light level is thus achieved as people arrive for work. However, this does cause an uneven usage of the lights. The switching patterns should therefore be rotated to give 'even wear'. To a large extent this is done in the ECS switching control at Lloyd's.

For the purposes of a case study one specific floor was selected namely floor 2. This floor contains 760 luminaires all under the control of ECS and was thus thought to be a good example. The lighting on floors 3 to 10 is similar with the exception that the number of luminaires decreases on the higher floors as the size of the floors decrease.

H2 IBMS Data

The CALMS computerised maintenance management system at Lloyd's records the service history and financial costing data on most of the plant equipment within the building. Breakdown data was obtained for every ECS controlled light fitting on floors 2 and 4 from April 1990 to January 1991. Each breakdown is entered on the CALMS database. Work dockets are then generated by CALMS and on each docket the fault is described along with the repair time and when it was completed. The faults were classified into one of five main groups shown in Table H1. The cause of the fault was either the starter, the tube, the circuit breaker or some combination of all three. Secondly, the fault data was summarised in the form of a distribution table for each maintenance period since the lamps were last block replaced. The number of faults in each period were totalled along with the number of hours of maintenance time. This enables the frequency distribution of the individual faults to be calculated along with the average repair time for each job. The graph in Fig.H1 shows the number of breakdowns per maintenance period which resulted in the changing of the tube.

No	Description	Code	Frequency	%	Av. Time (hours)
1	replace faulty lamp	1	54	54	0.72
2	starter switch faulty	2	3	3	0.58
3	faulty electrical supply	3	1	1	1.75
4	faulty lamp and starter	1,2	34	34	0.72
5	no fault	4	2	2	0.96
6	replace two luminaires	1,1	4	4	1.04
7	replace three luminaires	1,1,1	2	2	1.63

Table H1 Lighting Fault Classification

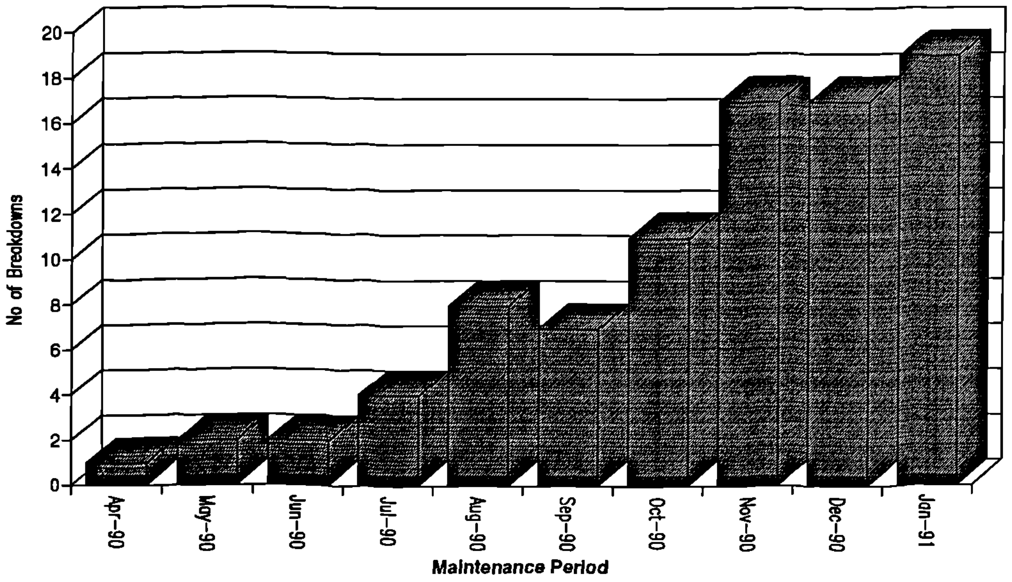


Fig.H1 Frequency Distribution of Breakdowns

It can be seen from table H1 that 88% of all lighting faults result in the changing of the tube (54% tube only and 38% tube and starter). It is interesting to note that on average it takes no extra time to change the starter at the same time as a tube replacement. This is what one would expect as most of the maintenance time is taken up by reaching the faulty lamp and dismantling the fitting.

Besides the breakdown data specific data on the light fittings is also required. This data such includes the material cost of each tube and starter, the average life and the labour rate for spot replacements. Lloyd's also has an additional fixed cost for every breakdown reported as each fault has to be entered on CALMS and then to process the docket. The cost per lamp of undertaking a block change, including cleaning of the fitting, was also obtained from CALMS data. A block replacement was last conducted at Lloyd's during the first quarter of 1991.

The switching patterns were obtained for the luminaires on floor 2 and are included in. The daily lighting cycle starts with a phased switch on at 5am in the morning. At this time 40% of the lights in the main gallery and the corridor are powered up. At 8am the rest of the gallery lights come on along with the perimeter lighting around the edge of the building. The lighting power down starts at 7pm with a 50% shut down followed by the rest of the lighting shut down at 7.30pm except for the perimeter lights. These are finally switched off at 11.30pm. The pattern is slightly more complicated than this as the two blocks of lights switched at 5am and 8.00am are interchanged daily in an attempt to ensure a more even wear on the fittings. This is very important because if some of the fittings are used significantly more than others this can effect the breakdown patterns. For example, the lights around the perimeter seem to have on average 6 more hours use a day than the other gallery lights. This works out at around 130 more hours a month and for an average life of around 8000 hours they would reach their average life 9 months before the rest of the fittings on the floor. Performing a block replacement at this time, simply because of excessive failures of the perimeter lighting, may be uneconomic as it would mean that 90% of the lighting on the floor may still have many months of useful life.

H3 Prediction

For effective optimisation of planned maintenance the condition of the lighting installation must be predicted at a specific time in the future. This prediction is based on the archived data on breakdown history and depreciation of light level.

The light level from a light fitting falls with time. This decrease in output is due to a combination of the two independent factors. Firstly the output from the tube itself will fall, very rapidly at the start of its life and then at a rate proportional to the amount of time it is used. The second factor which reduced light output is the build-up of dirt in the light fitting. This increases at a rate proportional to the elapsed time since the fitting was cleaned. In both cases the decrease in output can be approximated to a linear equation with the addition of an initial constant.

$$\% \text{ dirt losses} = I_d + I \times D_d$$
$$\% \text{ tube losses} = P_u / 1000 \times D_t + I_t$$

where,

D_d = depreciation rate per maintenance period due to dirt
I_d = initial depreciation due to dirt after the first period
D_t = depreciation rate of tube output per 1000 hours of use
I_t = initial depreciation in light from tube after first 1000 hours

%	Depreciation	Depreciation in output due to dirt	
	of tube	'clean' interiors	'dirty' interiors
Initial Depreciation (I)	0.05	0.02	0.06
Linear Depreciation (D)	0.02	0.01	0.02

Table H2 Luminaires Linear Depreciation Constants

Table H2 shows the values for the depreciation constants used[45]. Once the coefficients D_d, I_d, D_t, I_t are known the light level can be predicted at any given time if the elapsed time since the last cleaning and the rate of use per week are known. For a building such as Lloyd's the 'clean interior' constants are used.

T = proposed maintenance interval in months
U = monthly use in hours for a particular luminaires

For an individual lamp the predicted use after T maintenance periods is,

$$P_u = U \times T$$

and the build up of dirt on the fitting during this time and tube depreciation would cause,

$$\% \text{ dirt losses} = T \times D_d + I_d$$

$$\% \text{ tube losses} = P_u / 1000 \times D_t + I_t$$

This would cause the light level to fall to a minimum % level, L_m, after T months of,

---

[45] Source: Robinson W., Strange J., "'The Maintenance of Lighting Installations", Trans. of Illumination Engineering Soc. (London), Vol. 20 No.5

L_m= (1 - dirt losses) × (1 - tube losses) ..... (23)

which would mean an average % light level, L_{av}, over the T months of,

$L_{av} = 1 - [(1 - L_m) / 2]$ ..... (24)

and throughout this time an average % lost light of,

$L_l = 1 - L_{av}$ ..... (25)

By making spot measurements at specific times the constants D_d, I_d, D_t, I_t can be evaluated more accurately. These will change slightly from the approximate figures given in the IES table depending on the installation and usage patterns.

By taking light level readings L_0, L_1, L_2 at three separate times these figures can be calculated. L_1 is the light level reading at a time T_1 after installation and L_2 is the reading after a time T_2. The reading L_0 should be taken within one month of installing the new tube and cleaning the fitting. This can be used to calculate the initial depreciation constants.

T = proposed maintenance interval in months
H = Hours use at time after maintenance T periods

Reduction in light output = (1 - fraction of light lost from tube)
× (1 - fraction of light lost due to dirt)

$L_1 = [1 - (T_1 \times D_d + I_d)] \times [1 - (H_1 \times D_t + I_t)]$ ..... (26)

$L_2 = [1 - (T_2 \times D_d + I_d)] \times [1 - (H_2 \times D_t + I_t)]$ ..... (27)

This gives two simultaneous equations which can be solved to give D_d and D_t assuming I_d and I_t are found previously from L_0.

The data on reported breakdowns during each maintenance period shown in table H3 can be represented in the form of percentage luminaires surviving with respect to time as shown in Fig.H2. This graph represents the survival probability function for the life cycle. The data given on failures only covers the first part of the full graph as the fittings are block replaced well before the 50% failure point, as this is most economical. However, the survival probability can be predicted outside the known data range by extrapolation.

In the case of luminaires which exhibit a definite wear-out failure pattern the normal distribution provides the best fit. They tend to fail at some mean operating age, μ, with some failing sooner and some later, thus a dispersion, of standard deviation σ, in the recorded times to failure.

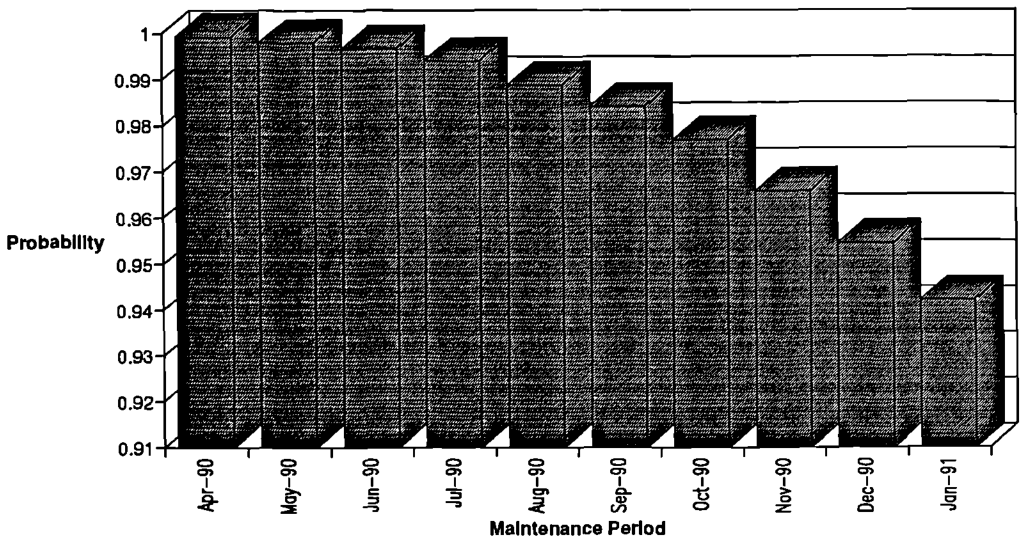


Fig.H2 Survival Probability of Luminaires

Period	Frequency	Cumulative	Probability
Apr-90	1	1	0.999342105
May-90	2	3	0.998026316
Jun-90	2	5	0.996710526
Jul-90	4	9	0.994078947
Aug-90	8	17	0.988815789
Sept90	7	24	0.984210526
Oct-90	11	35	0.976973684
Nov-90	17	52	0.965789474
Dec-90	17	69	0.954605263
Jan-91	19	88	0.942105263

Table H3 Lighting Breakdown Data

To be able to predict the number of failures at a particular time the cumulative distribution function is required. This probability can be obtained from the relevant probability density function as follows,

$$F(t) = \int_{-\alpha}^{t} f(t)\, dt = \frac{1}{\sigma\sqrt{2\pi}} \int_{-\alpha}^{t} \exp\left[\frac{-(1-\mu)^2}{2\sigma^2}\right] dt \quad \ldots\ldots \quad (28)$$

One simple way to represent this failure prediction function is by means of the function shown below,

Predicted failure probability,

$$P_f = \left\{ 1 \Big/ \left[1 + \left(\frac{P_{uh}}{A_{lh}} \right)^{i} \right] \right\} \quad \ldots\ldots \quad (29)$$

Predicted survival probability,

$$P_s = (1 - P_f) \quad \ldots\ldots \quad (30)$$

It is not exact but has the advantage that it can be represented by only two numeric constants. The first A_{lh} is some indication of the average life of the tube in hours which is the value around which the curve starts to fall off. This will differ slightly from the manufactures own data as the average life is to some extent a function of the switching pattern. The second constant, i , is a measurement of the rate of this fall off. The term P_{up} is the predicted use for the luminaires in hours.

Period	Period	Probability	Prediction	Error
Apr-90	16	0.999342105	0.9972049	0.002137
May-90	17	0.998026316	0.9957593	0.002267
Jun-90	18	0.996710526	0.9937219	0.002989
Jul-90	19	0.994078947	0.9909089	0.003170
Aug-90	20	0.988815789	0.9870982	0.001718
Sept90	21	0.984210526	0.9820270	0.002184
Oct-90	22	0.976973684	0.9753918	0.001582
Nov-90	23	0.965789474	0.9668513	0.001062
Dec-90	24	0.954605263	0.9560342	0.001429
Jan-91	25	0.942105263	0.9425519	0.000447

Table H4 - Lighting Failure Prediction Data

The actual data only covers up to about 40% of the luminaire's life and hence some form of extrapolation must be employed to estimate the average life time for these tubes. For the data given in table H4 a simple algorithm can be applied to generate these two values.

For the given data if,

$A_{lh} = 37.5$

$i = 6.9$

This function then gives the predicted values listed in H4.

The last column shows the error between the predicted and the actual data. This shows that the error in the predicted data is less than 0.5% for the known data range. The actual and predicted data is shown plotted in Fig H3.

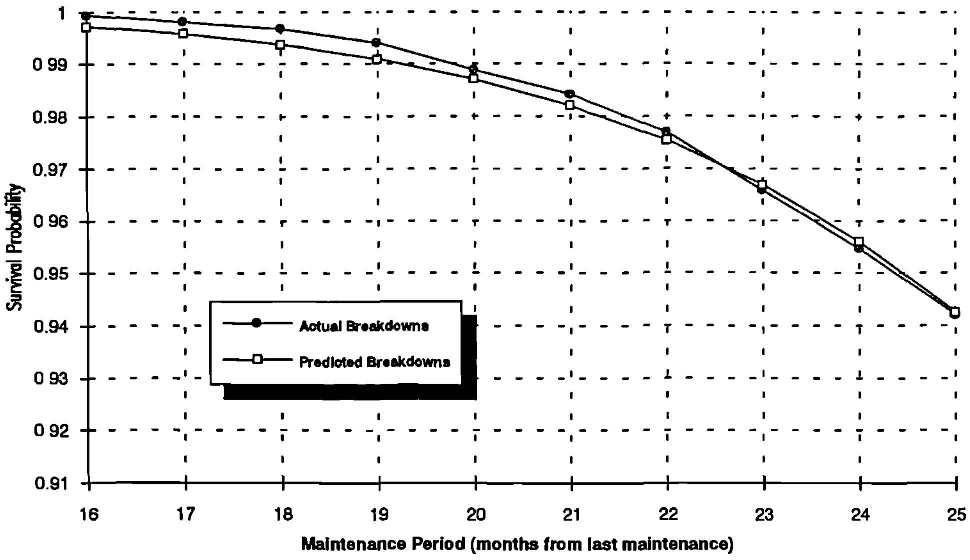


Fig H3 Fitting a Curve to Actual Breakdown Data

This function can now be used to extrapolate outside the known data range and predict the failure rate up to and beyond the average life of the luminaires. This enables the cost of breakdowns to be predicted up to and beyond the average life.

H4 Lloyd's Lighting Data

A KBS approach to lighting maintenance within an integrated building management system would improve maintenance management in the following four ways.

- Enable luminaire failures to be predicted based on historical breakdown data which would in turn enable maintenance costs both labour and materials to be predicted.
- Predict lighting levels for an installation at a given time based on actual light level readings. This may reduce the number of spot light readings that have to be made to assess the condition of the lighting installation prior to bulk lamp replacement.

- Advise on bulk maintenance based on economic considerations, occupant comfort and finite maintenance resources.
- Provide cost savings by administering a system of replacing spot failures with used lamps which have sufficient life left to last to the next bulk replacement.

The KBS itself would require minimum overheads as the data required for its analysis will be contained within the IBMS. At Lloyd's, this data is currently distributed between the CALMS and ECS systems but without a direct means of access.

The data supplied by Lloyd's was as follows,

n = 760 for Gallery 2 and 9800 for the whole building
C_{lb} = £8.73 per luminaires including cleaning
C_u = £8.70 (£7.50 for the lamp, £1.20 for the starter)
C_{ls} = £17.50 (£14.50 per hour plus FDA. charge for £3.00 per call)

The economic maintenance interval is predicted to be around 42 periods. This is after the lighting level has fallen to a min level of 40%. In this case the minimum light level criteria outlined in above will dominate the economic considerations and planed maintenance for this floor will be advisable in February 1993. For example, if the maintenance were to be delayed until May 1993 the theoretical increased cost overall to maintaining the building would be £1726.00 for Gallery 2 alone and possibly ten fold when considering the lighting in the building as a whole. This is an example of the sort of useful information that the KBS model of the building can calculate to assistant the staff in maintenance scheduling.

APPENDIX I - PUBLICATIONS

The following publications have so far resulted from this work:

[1] Clark G., Mehta P., "Optimised Maintenance management in Building Management Systems", 5th International Conference of Application of Artificial Intelligence at Padderborn, Germany, June 1992

[2] Clark G., Mehta P., Thomson T., "Intelligent Lighting Maintenance in large buildings", Universities Power Engineering Conference, Bath, September, 1992

[3] Mehta P., Clark G., Thomson T., "Intelligent Buildings: into the future", IEEIE Journal of ElectroTechnology, Institution of Electrical and Electronic Incorporate Engineers Journal, July 1992

[4] Clark G., Mehta P., T Thomson, "Occupancy Prediction within Energy Management Systems", International Conference on Building Design Technology and Occupant Well-Being, Palais Des Congres, Brussels, February 1993

[5] Clark G., Mehta P., "A Rule-Based Approach to Load Control in Large Buildings", UPEC '93, Staffordshire University, September, 1993

[6] Clark G., "Sick Building Syndrome - Who is to Blame?", Invited paper for: Building Services Colloquium, T Dunwoody & Partners, Kenton, London, October 1993

[7] G Clark, P Mehta, R Prowse, "Knowledge-Based Elevator Controller", IEE Control '94, University of Warwick, March 1994

[8] G Clark, P Mehta, 'Intelligent' Integrated Building Management Systems, Intelligent Buildings, Tel Aviv, Israel, January 1995

[9] G Clark, P Mehta, IBMS Assisted Facilities Management,.Intelligent Buildings, Tel Aviv, Israel, January 1995

REFERENCES

1. Rigg D. W., "Intelligent Buildings - the Japanese approach, High Tech Buildings 1988", Conference Proceedings, 1988
2 Status Report: "Intelligent Office Building Market in Japan", Intelligent Buildings Quarterly, Vol.2 No.3 1992
3. Pengilly R., "Does the future of high-tech buildings depend on our ability to manage them?", Article, High technology Buildings 1987
4. Gregory D P, "Adaptive Building Envelopes", Technical Note TN3/86, Building Services Research and Information Association, 1989
5. Battle G., "Energy : the art of architecture", Building Services Journal, Dec. 1990
6. Dubin F. S., "Energy Efficient Building Design, Innovative HVAC, Lighting, Energy Management, Control and Fenestration", Applied Energy 1990 ,Vol.36 No.1-2 pp.11-20
7. Kreager P. S., "The Intelligent Building Telecommunications Infrastructure", IEEE Communications Magazine, Vol 29, Iss 4, pp 42-48, 1991
8. Heating and Ventilation Contractor's Association, "Standard Maintenance Specification for Mechanical Services in Buildings - Control, Energy and Building Management Systems", Vol.3, 1992, TH6025 S84
9. Terry R., Weaver P.E., "The Integrated Building Automation System", The intelligent building source book, Fairmont press, ISBN 0-88173-019-X
10. Scheepers H. P., "Inside building management systems, manual and supporting systems", ISBN 90-72743-02-4
11. Davies C. R., "Computer-based planned maintenance programmes", Property Management 1989/1990, Vol.8 No.1 p40-60
12. Armstrong C. N., "Selecting a Computerised Maintenance Management Program", Pulp & Paper-Canada Vol. 89. Iss 11 pp 36-39 1988
13. Finley M.R., Karakura A, Nbogni R., "Survey of Intelligent Building Concepts", IEEE Communications Magazine, Vol. 29, Iss. 4, pp 18-23, 1991
14. Birtles A. B., John R W Smith J T, "Performance of a trial energy management system", Building Research Establishment Information Paper IP 2/85
15. Atkinson G., "Does it pay to save it?", Building Services, April, 1985
16. Robbens E.G., "BMS Dilemmas for system operators", IEE Building Management Systems Colloquium, Computing and Control Division, Savoy Place, London, March 1992
17. DeBartolomeis F., "Computer integrated buildings : a design approach", High tech Buildings 1988
18. Dymott J., "Don't miss the bus", Building Services Dec. 1990

19. Bushby S., Newman H., "Standardising EMCS Communication Protocols", ASHRAE Journal American Society of Heating Refridgerating and Air Conditioning Engineers, 1989 31 No-1 p33
20. J Wilkins, "What are smart sensors?", Building Services, Dec. 1990
21. "Intelligent Sensors", Control and Instrumentation, Vol 21 No 11, November 1989 et seq.
22. "Facilities Management", Building services, August 1991, p32
23. Rex Pengilly, "What integrated services need to keep working", High Tech Buildings 1988
24. Porter B., Jones H., McKuewn C., "Real-time expert tuners for PI controllers", IEE Proceedings, Vol. 134, Part D, No 4, July, 1987
25. Shaw M., "Making Intelligent Buildings a reality", High-Tech Buildings Conference, 1989
26. Nolden C., "Predictive Maintenance", Plant Engineering Feb 1987, Vol.41 No.4 p38-43
27. Ashley S., "Maintenance & Refurbishment - Why the Jungle?", Building Services Feb. 1985
28. Terry R., Weaver P. E., "Integrated Maintenance Management", The intelligent building source book, p65, Fairmont press, ISBN 0-88173-019-X
29. Egol L., "Computerised Maintenance Management Saves Money", Chemical Engineering, Vol. 95 Iss 13 pp 107-109 1988
30. Egol L., "Save Dollars on Maintenance Management", Chemical Engineering, Vol. 97 Iss 6 pp 157 1990
31. Robert M., "Contracting out Maintenance", Building Services, October 1989
32. Kelly A., Harris M. J., "Management of Industrial Maintenance", Butterworths, 1983, TS192 K45

33 CORMAC Systems (Europe) Ltd Sales Literature, 18 High Street, Bagshot, Surrey, GU19 5AA

34. Sedacca B., "Experts on Screen", Building Services, August 1991, p32
35. Shaw M., "Expert systems and BEMS", Energy Efficiency Office, Building Research Establishment, Information Leaflet 21, July 1990
36. Willis S., "Technical file - Helping to make sense of BMS", Building Services August 1991, p43
37. The search for knowledge, Tomorrow's ideas today, 1987 CIBSE Technical Conference
38. Finley H. F., "Reduce Costs with Knowledge-Based Maintenance, Hydrocarbon Processing", Vol. 66 Part 1 p64-55 Jan. 1987

39. Culp C. H., "Expert Systems in Preventative Maintenance and Diagnostics", ASHRAE Journal -American Society of Heating Refrigerating and Air-Conditioning Engineers, 1989 Vol. 31 No.8 p24-27,
40. Dohrmann D. R., Alereza T., "Analysis of survey data on HVAC maintenance costs", Dirt and Maintenance, p27, ASHRAE Transactions,Vol.92 part 2A p550-565
41. Pignataro V., "Details of the Computerised Maintenance Management System at Quebec and Ontario Paper Co. Ltd Thorold Ontario", Pulp & Paper-Canada, Vol. 90, Iss 1, pp109-110, 1989
42. Finn G. A., Reinschmidt K. F., "Expert Systems in Operation and Maintenance", Chemical Engineering 1990, Vol.97 No.2 p.131
43. Warren T., Pender R., "Maintenance Management Control Through Utilisation of a Computerised System", CIM Bulletin, Vol. 81 Issue 911 pp 94-94 1988
44. Schneider W., "Maintenance Management Control Through Utilisation of a Computerised System At the Key Lake Project", CIM Bulletin, Vol. 83, Iss 942, pp 67-71, 1990
45. Arueti S., Okrent D., "A Knowledge-Based Prototype for Optimisation of Preventative Maintenance Scheduling", Reliability Engineering and Safety, Vol. 30, Pt 1-3, p93-114, 1990
46. Majstorovic V. D., Milacic V. R., "Expert Systems for Maintenance in the CIM Concept", Computers in Industry, Vol.15 No.1-2 pp.83-93, 1990
47. Bedewy M. K., Sherif A. O., Abdallah A. Y., "An Interactive Computer Program for Maintenance Systems", Reliability Engineering & System Safety 1989,Vol.25 No.4 pp.349-364
48. Majstorvic V. D., "Expert Systems for Diagnosis and Maintenance", The State-Of-The--Art, Computers in Industry 1990, Vol.15 No.1-2 pp.43-68
49. Purcell E., "Fault-Diagnosis Assistant", IEEE Circuits and Devices, 1988 Vol. 4 No 1 p47-59
50. Wang X., Coke R., "Optimal Inspection Sequence in Fault-Diagnosis", Reliability Engineering & System Safety, 1992 Vol. 37, No 3, p207-210
51. Cardozo E., Talukdar S., "A Distributed Expert System for Fault-Diagnosis", IEEE Transactions on Power Systems, 1988 Vol. 3 No 2 p641-646
52. Bezanov G., Mehta P., Thomson T., "Application of expert systems to fault diagnosis of electrical power networks in modern buildings", 27th Universities Power Engineering Conference, Bath, September 1992 paper
53. Pet T., Dhurja P., "Object-Based Automated Fault-Diagnosis", Chemical Engineering Communications, 1991 Vol. 102 p107-126
54. Yoon W., Hammer J., "Deep Reasoning Fault Diagnosis - an Aid and Model", IEEE Transactions of Systems Man and Cybernetics, 1988 Vol. 18 No 4 p659-676

55. Bublin S., Kashyap R., "Generating Fault Hypotheses with a Functional-Model in Machine-Fault Diagnosis", Applied Artificial Intelligence, 1992 Vol. 6, No 3 p353-382
56. Wang X., Dillon T., "A 2nd Generation Expert System for Fault-Diagnosis", International Journal of Electrical Power & Energy Systems, 1992 Vol. 14 no2-3 p212-216
57. Specht D., Weiß S., "Supporting Model-Based Diagnosis with Explanation-Based Learning and Analogical Inferences", Proceedings of the 5th International Conference on Industrial and Engineering Applications of Artificial Intelligence and Expert Systems, Paderborn, Germany, June 1992
58. Sorsa T., Koi H., Koivisto H., "Neural Networks in Process Fault Diagnosis", IEEE Transactions on Systems Man and Cybernetics, 1991
59. Ray A., "Equipment Fault-Diagnosis - A Neural Network Approach", Computers in Industry, 1991, Vol. 16 No 2 p169-177
60. Iordache O., Corriou J., Tondeur D., "Neural Network for System Classification and Process Fault-Detection", Hungarian Journal of Industrial Chemistry, 1992 Vol. 19, No 4, p265-274
61. Kurmar R., "Equipment fault diagnosis A neural network approach", Computers in industry, Vol. 16, 1991, pp.169-177
62. Ungar L., Powell A., Kamens S., "Adaptive Networks for Fault Diagnosis and Process Control", Computers & Chemical Engineering, 1990 Vol. 14 no 1-4 p561-572
63. "Dutch cure for sick building syndrome", Building, Vol.7, Sept. 1990, p19
64. Sakamura K., "The Tron Building", Project Report on The Real Time Operating System Nucleus (TRON), Tokyo University, Japan, September 1992.
65. Fazio P., Zmeureanu R., Kowalski A., "Select HVAC - Knowledge-Based System as an advisor to configure HVAC systems", Computer-Aided Design, 1989 21 No-2 p79-96
66. Akagi S., Fujita K., "Building an Expert System for Engineering Design Based on the Object-oriented Knowledge Representation Concept", Journal of Mechanical Design 1990, Vol.112 No.2 pp.215-222
67. Dubin F., "Energy Efficient Building Design - Innovative HVAC Lighting- Energy Management Control and Fenestration", Applied Energy Engineering - Transactions of the ASME, 1990 Vol. 36 N 1-2 p11-20
68. Bernard C., Guerrier B., Rossetlouerat M., "Optimal Building Energy Management (Modelling and Control)", Journal of Solar Energy Engineering - Transactions of the ASME, 1992 Vol. 114 No.1 p2-22
69. Clark G., Mehta P.A., Thomson T., "Intelligent Buildings - Into the Future", IEEIE Journal 1992 Vol.3 No.3

70. Akagi S., Fujita K., "Building an Expert System for Engineering Design Based on the Object-oriented Knowledge Representation Concept", Journal of Mechanical Design, Vol.112, No.2, 1990, pp.215-222
71. Won K., "Object-Oriented Databases: Definition and Research Directions", IEEE Transactions on knowledge and data engineering, Vol. 2 No.3 September 1990
72. Mcfadden R.H., "Developing a Database for a Reliability, availability and maintainability improvement program for industrial plant or commercial building", IEEE Transactions on Industry Applications, Vol. 26 Iss 4 pp 735-740 1990
73. Rumbaugh J., "Object-Oriented Modelling and Design", 1991, Prentice Hall, ISBN 0-13-629841-9, QA76.64.025
74. Neuron Data Inc., "Nexpert Object Reference Manual", January 1991, Neuron Data, 156 University Avenue, Palo Alto, California 94301.
75. Reddy-TA Norford-LK Kempton-W, "Shaving Residential Air-Conditioner Electricity Peaks by Intelligent Use of the Building Thermal Mass", ENERGY,Vol 16, Iss 7, 1991, p1001
76. Dettmer R., "Meter Power", IEE Review, June 1992
77. Wong K, "Algorithm for Load-Shedding Operations in Reduced Generation Periods", IEE Proceedings C, Generation Transmission and Distribution, 1992, Vol.139, No 6, p478-490
78. Adibi M., Thorne D., "Local Load Shedding, IEEE Transactions on Power Delivery", 1992 Vol. 7, No 2 p480-488
79. Shah S., Shahidehpour S., "A Heuristic Approach to Load Shedding Scheme", IEEE Transactions on Power Systems, 1989 Vol. 4, No 4, p1421-1429
80. Fox L., Mayers D. F., "Computing methods for scientists and engineers", Oxford Press, QA297
81. McLaughlin R.K., "Heating Services Design", 1981, Butterworths, TH7222 M22
82 O'Callaghan P., "Energy Management", 1992, McGraw-Hill
83. Heath S., "Echelon - Networking Control", IEE Review, October 1992
84. Gabor A., Barna G.,"Knowledge-Based System for Supporting Statistical Database Management", KBS Journal Vol. 1 No 5, December 1988
85. Jardine A. K., "Maintenance Replacement and Reliability", Department of Engineering Production, Pitman Publishing, 1973
86. Jirousek R., "Methods used in Probabilistic Expert Systems for Knowledge Integration", 1989 AI Conference Prague
87. Lutcha J., Zejda J., "Knowledge Represented By Mathematical Models for Fault Diagnosis in Chemical Processing Units", Knowledge-based Systems, March 1990
88. Gertsbakh I., "Models of Preventative Maintenance", Volume 23, Studies in mathematical and Managerial Economics, North Holland Publishing Company, 1977

89. Engert C., "Systems and the Management of Buildings, A User Case Study", Presentation given by the Estates Manager of the Lloyd's 1986 Building
90. Prowse R., Thomson T., Howell D., "Design and Control of Lift Systems Using Expert Systems and Traffic Sensing", Proceedings of ELEVCON '92, Amsterdam, The Netherlands.
91. André P., Nicolas J., "An Integrated Software for Building Design", International Conference on Building Design Technology and Occupant Well-being in Temperate Climates
92. AS-1 User's Guide, CAOS V4.03 Update Guide, Cambridge Research Systems, C/o Fairchild Ltd, 78-80 Portswood Road, Southampton, UK, SO2 1FW
93. Rumelhart D.E., McClelland J.L., "Parallel Distributed Processing", MIT Press, Cambridge, 1986.